Learning with interpretable models

Habilitation thesis
(cumulative)

Dr. rer. nat. Frank-Michael Schleif

To my parents and Ute

Contents

1 Introduction **1**

2 Classification with uncertainty **15**
2.1 Fuzzy Classification by Fuzzy Labeled Neural Gas 15
2.2 Spectral data analysis by Fuzzy Self Organizing Maps 31
2.3 Prototype based Fuzzy Classification in Clinical Proteomics 49

3 Improved evaluation, interpretation and domain knowledge integration **67**
3.1 Cancer Informatics by Prototype-networks in Mass Spectrometry 67
3.2 Supervised data analysis and reliability estimation for spectral data 87
3.3 Evolving trees for mass spectrometry data 109
3.4 Genetic algorithm employing metric learning for NMR data analysis 127

4 Large scale models **149**
4.1 Efficient kernelized prototype based classification 149
4.2 Linear time relational prototype based learning 167

5 Acknowledgment **185**

Chapter 1

Introduction

Interpretable models

The current decades in the field of computer science and machine learning are dominated by an increasing amount of electronically available data also known as the *big data* challenge [75, 38].

With projects like the genome sequencing initiative, huge amounts of sequence data are available, social media generate continuously large amounts of data in different formats, and many other sources provide data and challenging data analysis and interpretation tasks [36, 82, 43, 68, 6, 79].

The interpretation of these data and its structuring in form of compact models is widely considered as important to access these data in an efficient manner.

To get access to these sources, efficient data analysis algorithms and models are necessary. One can consider an algorithm to be efficient, if it successfully tackles the problem and provides a model at acceptable costs with accurate results and good generalization ability on new data. The costs can be defined by means of the necessary training time, by the amount of memory necessary to obtain and store the model, but also by evaluating the time a model needs to give a prediction on a new item.

The problem of large scale data analysis has been addressed already by different approximation or sparsity strategies, heuristics or by novel mathematical concepts exploring e.g. geometric properties of the data space, or by incorporating additional domain knowledge simplifying the original problem [59, 47, 111, 121, 12]. Also the extraction of simplifying rules from more complex models has been discussed [42, 56]. Here multiple disciplines and research fields often have to work together in an interdisciplinary way to define appropriate solutions. While the domain expert, may be e.g. a biologist or clinician, the measurement system is best understood by a physicist and all have to work together with mathematicians or computer scientists to define an efficient algorithm for the analysis of the data.

This collaboration and communication can only be efficiently achieved by the use of interpretable models. As recently discussed in [11] there are three major requirements for those models: (1) they have to map onto the domain knowledge, (2) should ensure safe operations across the full operational range of model inputs and (3) should accurately model non-linear effects.

These requirements provide the domain expert with some guarantees to make the model valuable but also permit to communicate the modeling behavior and the decisions of the model to the expert in an accessible way. Multiple traditional and state of the art algorithms

behave like black box approaches. Prominent examples are classical neural networks [60, 45] but also kernel machines like the Support Vector Machine (SVM) [122, 108]. Although these methods are very effective and successfully applied in many fields it is often complicated to adapt these algorithms to new problems and their behavior is often hard to predict, especially in the case of errors and unexpected input data. In the field of kernel machines the choice of the kernel and its parameters is widely open and addressed either by ad-hoc decisions or by meta-evaluation schemes. Often the domain expert is not even aware about additional constraints of the model, like positive semi-definiteness.

Principal component analysis (PCA) [67] constitutes one very simple example of a widely interpretable model. In general it used as a tool to reduce the dimensionality of the input data. Dimensionality reduction is often a useful step to simplify the given problem, to ease the subsequent data analysis and to make the results easier to interpret. However it is also a critical step which may reduce the expressive power of the data. It is extremely popular in the field of life sciences, but often used a bit careless. The model outputs of a PCA are quite accessible because the principle components are linear combinations of the original points as reflected in the loadings. In this way the PCA is an interpretable model with respect to point (1). The expert knows the meaning of the original input features and may also be able to draw conclusions about the combination of features or their omission. However, it is much more complicated to explain how these components, or precisely the loadings, are obtained and which limitations are behind the model generating algorithm. Inherently the PCA covers linear effects, so fails to address point (3) and for non-linearities in the data more complex approaches like kernel-PCA [105], the Self-Organizing-Map (SOM) [74] or other types of non-linear projection methods [78] are available. Also the effect of outliers, addressing the priorly mentioned point (2) of a rather uncommon range of model inputs is not easy to communicate and may invalidate the results.

The mapping of the model into the domain knowledge (1) can be addressed in very different ways. Considering e.g. decision trees [58], the model, given as a tree, often reflects the decision process of the domain expert and additionally can be visualized with annotations to communicate the decision of the model. An even more complex model of this type can be found by Bayes networks [19] which try to cover the probabilistic dependencies between different objects and are also often visualized for a better communication with the domain expert. Obviously the internal representation of the data by the model generating algorithm is a key element of each learning approach and a anchor to incorporate domain knowledge. Recently the concept of relevance learning or matrix learning [57, 104] has been proposed to incorporate domain knowledge or auxiliary information. Both strategies are metric adaptation schemes and are still sufficiently simple to be communicated to the expert and to provide sufficient information to link the model output with the original data [65, 4].

In the following we will consider different questions regarding the modeling of *prototype-based learning algorithms* and how these models can be extended to support domain knowledge integration and to provide interfaces for model and data interpretation. Prototype based models form a specific family of algorithms and can be considered as another primal example for interpretable models.

Their main characteristic is a typically simple and sparse model, constructed from a subset of the original data points. Accordingly all these models take the necessary prerequisites of interpretable models into account by: (1) representing the data by prototypes which itself look like data, (2) building their functionality on the distances of data points and prototypes, whereby prototypes are located in safe regions of the data space, (3) allowing non-linear effects due to the tessellation of the space and the corresponding function in terms of prototypes.

As outlined in the following, multiple extensions of this principle can be provided, approaching different data analysis challenges. Due to the inherent connection to the original data space these models are traditional candidates for interpretable models, often accompanied by a rigid mathematical framework and theoretical guarantees [55, 17, 101, 16].

As will be outlined in the following more advanced prototype approaches provide not only good interpretability but are also sufficiently robust and flexible to address the three major points of interpretable models. We will also discuss strategies to improve the efficiency of these algorithms.

We will now first shortly introduce the standard prototype models and motivate algorithmic questions arising in the context of these models which are to be answered to justify their use as interpretable data analysis approaches. Then we give a short overview of important or recent results in the literature. Finally, nine articles are included which each constitute a major contribution to a different question within this context.

Prototype based learning

Prototype based learning is an interesting strategy to define efficient data analysis algorithms. It has been successfully used e.g. in the field of clustering [59], (semi-) supervised-learning [100, 23] and data embedding [51, 48].

A common principle of these algorithms is the definition of the models by means of *few*, so called, *prototypes* or representatives of the original or some derived data space as formalized in the subsequent articles. Basically we consider a vector space with vectors $\mathbf{x} \in \mathcal{R}^D$ providing a dataset $V \subset \mathcal{R}^D$, with D the number of input dimensions or features and $N = |V|$ as the number of samples. The general objective is to identify prototypes $\mathbf{w} \in \mathcal{R}^D$ representing V.

The details about the corresponding training procedures and recent extensions of classical concepts are discussed in more detail in Chapter 4, here we will simply assume that some training procedure exists to provide a prototype representation for V. The learning of these models can be done unsupervised, supervised or semi-supervised.

Classical techniques for the unsupervised setting are k-means, or for supervised problems learning vector quantization (LVQ) [74]. In the later case each vector $\mathbf{x}$ has an additional label $l \in \mathcal{L} \subset \mathcal{N}$ with $L = |\mathcal{L}|$ as the number of labels. In Chapter 2 this concept is generalized to vectorial label representations such that $l \in \mathcal{R}^L$ with the general constraint $\sum_i^L l_i = 1$. This setting is also called fuzzy-labeling and can be applied to the data, the prototypes or both.

The identified prototypes introduce a tessellation of the underlying data-space into different regions. In most cases this tessellation is disjoint such that a data-point $\mathbf{x}$ belongs only to a single prototype $\mathbf{w}$ according to some mapping rule. In general the winner takes all rule is used:

$$\mathbf{x} \mapsto \mathbf{w}_j \text{ where } d(\mathbf{x}, \mathbf{w}_j) \text{ is minimum} \tag{1.1}$$

with some distance measure, e.g. the squared Euclidean distance $d(\mathbf{x}, \mathbf{w}_j) = \|\mathbf{x} - \mathbf{w}_j\|^2$, breaking ties arbitrarily. While this is very common, more general cases using e.g. the softmax mapping criterion can be used leading to soft learning vector quantization schemes as shown e.g. in [101, 106] or more recently in a full probabilistic setting [102, 22].

Prototype based learning typically occurs in one of two principles, namely online learning, processing the data in an incremental manner, or using batch-learning where it is assumed that all data are available and used in common for an optimization step. More recently also streaming data have been considered within prototype-based learning schemes [2] the different concepts are used and explained in the subsequent chapters.

1	Unsafe label representation	Chapter 2
2	Insufficient supervised prediction	Chapter 3
3	Detection of novel inputs	Chapter 3
4	Insufficient model representation	Chapter 3
5	Prototypes from mixed data	Chapter 3
6	Limited interpretability of proximity models	Chapter 4
7	Limited scalability of proximity models	Chapter 4

Table 1.1: Challenges for prototype learning algorithms

For prototype learning techniques or especially LVQ techniques the solutions are represented by a small number of representative prototypes which constitute members of the input space. As a consequence the model, given by means of the prototypes can be inspected in the same way as the individual training data. Since the dimensionality of the points $\mathbf{x}$ is typically high, this inspection is often problem dependent: images, for example, lend itself to a direct visualization, oscillations can be addressed via sonification, spectra can be inspected as a graph which displays frequency versus intensity. Moreover, a low-dimensional projection of the data and prototypes by means of a nonlinear dimensionality reduction technique offers the possibility to inspect the overall shape of the data set and classifier, independent of the application domain see e.g [27, 98].

Prototype based learning algorithm define predictive models, which represent the data V by few examples and the decision function is often very simple. As mentioned before and detailed in the following, prototype learners are very flexible. Some standard approaches are however not fully interpretable models by means of the prior criteria and show limitations in the integration of domain knowledge or are less interpretable or accurate regarding the model output. Some more recent extensions of prototype approaches e.g. to address nonlinearity in the data, become quite complex and are also less interpretable. In this thesis multiple concepts are presented to overcome some of these problems and to bring prototype learning approaches closer to interpretable models in these settings.

In Table 1.1 some major problems of prototype learning algorithms and where they are address in this thesis are summarized:

- Problem 1 refers to a supervised learning problem and the special case where the given data are not labeled by a unique class assignment but are associated to multiple classes to some degree. This problem occurs also for the prototype, because their receptive field may not map to a single class.

- Problem 2 deals with the decision rule of the prototype classifier for supervised problems. Most often the winner takes all rule is used and the predicted label is the one of the closest prototype. This is a very coarse view on the tessellation of the data space by the prototypes, ignoring the proximity of the test item to the prototype and even more important the similarity of the new test item to the points of the receptive field or those which are nearby. The method of conformal prediction [107] offers interesting strategies to provide calibrated p-values and to enhance the label prediction by a measure of confidence and credibility. It is shown how this concept can be integrated also to prototype based learning approaches.

- Problem 3 tackles the problem of novel data given to a prototype model and is linked to problem 2. If a test item shows a data characteristic which was never seen by the

model it would be appropriate to report this point as an outlier or novel concept. It is shown that strategies from problem 2 can be used to approach this scenario.

- Problem 4 considers a specific problem in unsupervised learning but may also be of wider interest. A known prototype method for tree generation is formalized and concepts for domain knowledge integration and improved interpretation of the tree representation are provided.

- Problem 5 considers the special case where the prototype, as the model output, have to be identified from mixed data. In a specific scenario the usage of domain driven metric adaptation is discussed to identify the prototypes.

- Problem 6 occurs for prototype models generated on proximity data. Here the prototypes are generated as an indirect linear combination of the original data. These models are often quite dense and hard to interpret. Different strategies to simplify these models by sparsity and approximation methods are discussed. The final model becomes interpretable again.

- Problem 7 is again a major problem for prototype approaches of proximity data. The underlying data representation scales quadratic in the number of points which becomes prohibitive for larger data sets. Approximation strategies are discussed to keep prototype models applicable for larger scale problems.

Interpretability in prototype based models

As already mentioned prototype based learning often provides properties which permit interpretability or open ways to obtain models which are closer to generic interpretable models. Subsequently different concepts are discussed which are used to enable interpretable prototype based models.

Interpretability is interesting in virtually every domain and of special interest in problems where human inspection of the models or the model predictions is necessary [87]. The life science domain is an immediate application field often requesting for interpretable models, probably most pronounced in medicine [65, 4, 1, 80], but interpretable models are of very wide interest [25, 115, 8, 91].

Prototype learning not only permits the integration of domain knowledge, but also simplifies this step by providing direct access to the model parameters, especially the prototypes and also the metric parameters. This eases the design stage of a new algorithm since the intermediate approach and models are easier to communicate to the domain expert.

To make it a bit less abstracts lets consider a brief example illustrated in Figure 1.1. Here the objective was to learn a discriminative model for a large set of bacteria mass spectra fingerprints. Different bacteria cultures were measured by mass spectrometry, leading to finger-print spectra. They have been collected in a database and traditionally a new, unknown spectrum is compared to the whole database using a domain specific similarity measure. The prototype model provides a simplified representation of this database by few examples, leading to faster identification results. Additionally, and even more important in the practical application the model has to be interpretable to identify matching / non-matching regions in the test spectrum with respect to the model as shown in Figure 1.1.

A prototype model provides much more insight into the identification process and the model is more appreciated by the domain expert than standard black box approaches. A hierarchical approach focusing on the same problem is discussed in Chapter 3.

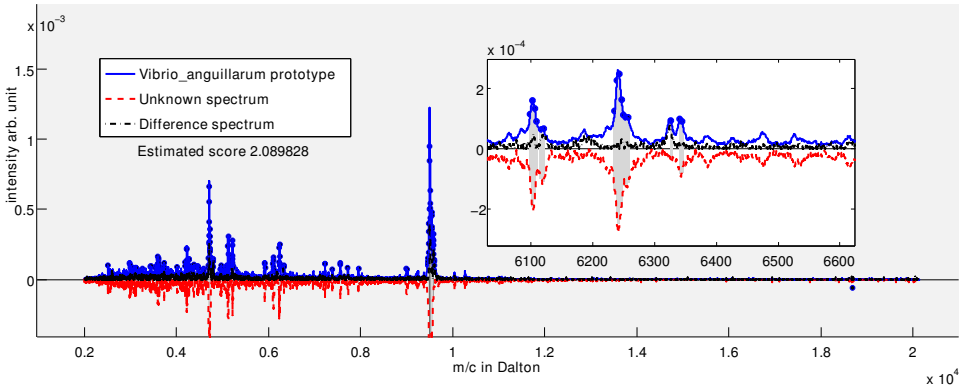


Figure 1.1: The prototype (straight line) represents the class of the test spectrum (dashed line). The prototype is labeled as *Vibrio Anguillarum*. It shows high symmetry to the test spectrum and the similarity of matched peaks (zoom in) highlights good agreement by bright gray shades, indicating the local error of the match. This is only meaningful if the identified prototype mimics the data characteristics of the represented class and the signal shape of the prototype is biochemically plausible. It allows to judge the identification accuracy using a domain specific bioinformatic score of the test spectrum. The score of > 2 indicates a good match. The prototype model allows direct identification and scoring of matched and unmatched peaks, which can be assigned to its mass to charge (m/c) positions, for further biochemical analysis.

The wide success of prototype approaches is reflected by a huge amount of publications and applications of prototype approaches in different domains see e.g. [74]. More recent work has shown the efficiency of prototype based learning also by theoretical results [13, 55, 101] providing e.g. generalization bounds for supervised learning.

Interpretability of data analysis models is also relevant in times of big data. While we may assume that fully automatic procedures are needed to deal with the data flood, a valid model often requests deep knowledge of the domain problems and this knowledge has to be transferred into the data analysis algorithm, accordingly interpretable models appear to be a good choice. Prototype approaches have been found to be flexible enough to integrate domain knowledge in different ways:

- flexibility regarding the used metric
- learning the parameters of the metric
- a specific optimization strategy e.g. taking class relations into account
- representing the prototypes or the model in an appropriate data structure or data format
- to learn representative features from the data
- enforcing sparse models

As discussed before the concept of relevance learning or metric adaptation was derived for supervised prototype learning to adapt the data representation such that domain specific or auxiliary information, like class information is effectively used [57]. Beside the priorly

mentioned aspects, metric adaptation is also a great strategy to improve not only the model but also to obtain better interpretability.

Metric learning has been used for global, local and class-wise metrics and to adapt the parameters of diagonal or mahalanobis-type metrics [104]. The analysis of the corresponding relevance profiles either locally, focusing on e.g. discriminating aspect of subgroups of the data or globally, explaining relevant input dimensions of the whole data set are very interesting for the typical domain expert. They can be plotted easily and can also be used for post analysis and processing steps like feature filtering strategies or even rule induction by considering individual relevance parameters as branching indicators [56].

Also very different types of metrics have been integrated as already discussed and some of them are very domain specific like for functional or taxonomic data [109] more details can be found in Chapter 3.

The dependency between classes, e.g. by an ordinal ordering was approached in [44] for prototype based learning, showing how to integrate domain knowledge by adapting the optimization scheme. The batch variant of the soft competitive learning (SCL) algorithm [83] effectively optimized the encoding of video streams [130], employing domain knowledge about the manifold of meshes.

Also domain specific data formats like complex-valued data [31], functional data [69], structured data and graphs [54, 81, 53, 39, 24] or similarity [89, 94, 35, 70] and dissimilarity data [37, 52] can be analyzed by prototype based models as shown in Chapter 3 and Chapter 4.

Learning of appropriate features or encodings from the data has also been approached using prototype learners with a good overview given in [63, 77] where also sparsity concepts are discussed [76]. Sparsity is one of the natural strategies to improve interpretability. Affinity propagation [49, 46] is a sparse prototype based model where the prototypes are exemplars from the original data set, leading to natural sparse models.

Sparsity concepts were also explored for prototype learners dedicated to similarity or dissimilarity data as shown in more detail in Chapter 4. For those methods the prototypes are represented as linear combinations of the original data and a *sparse* linear model is of interest using classical sparsity constraints.

We now briefly review major areas of prototype learning in relation to the priorly addressed problem fields. While the specific subsequent articles, approaching the mentioned problems, are widely self contained the following introduction will give a broader overview.

Unsupervised models

For unsupervised prototype-based learning the most prominent techniques can be distinguished into approaches with topological constraints or without such restrictions.

Approaches without topological constraints are the well known k-means clustering or soft-competitive learning, referred to as (SCL or NG) [83]. As already discussed before the idea is to cluster a large set of N data points by a small number of prototypes. Some unsupervised learning concepts are discussed in Chapter 4 and in a context of fuzzy labels in Chapter 2. Clustering is an extremely active field of research and has been addressed from different theoretical and practical perspectives [66]. Recently the focus has shifted to large scale issues and non-standard as well as structured data problems [35, 66]. In this work clustering is mainly considered for the clustering of similarity and dissimilarity data, without topological constraints, and a focus is given to large scale problems for this type of data.

The best known unsupervised prototype-learners with topological constraints are probably generative topographic mapping [20] (GTM) or the self-organizing-map (SOM) e.g.

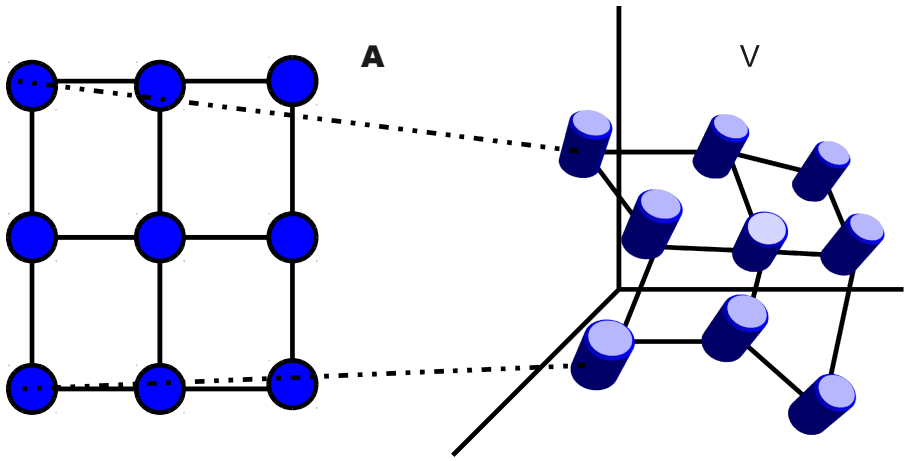


Figure 1.2: Schematic view of a topographic mapping e.g. by SOM or GTM. **A** defines the low dimensional (here $2D$) grid space and **V** the high-dimensional (here $3D$) data space. Prototypes are shown as circles/cylinders and two related prototypes are exemplarily connected.

in the variant of Heskes equipped with an appropriate cost-function [62]. The basic idea for this type of methods is to constrain the clustering model leading to a low-dimensional representation of the data on a grid. The representation is often chosen $2D$ or $3D$ on a rectangular grid. This grid provides a low-dimensional latent space which is fixed and the prototypes spanning the grid are non-linearly mapped into the high dimensional data space. The general problem is illustrated in Figure 1.2. This type of constrained clustering shows interesting properties given that the mapping from the latent space A to the data space V has been done in a topology preserving way. If the data are intrinsically low dimensional, neighborhood relations in the obtained low-dimensional map transfer to the high-dimensional data as well. The otherwise inaccessible high-dimensional data become interpretable in this way. This is obviously only valid in case of topology preservation; mathematical measures are provided in the literature to test the validity of the mapping [127]. Interestingly many real life problems can be approached in this way and are intrinsically low dimensional. This property is also used in Chapter 2 for a fuzzy variant of the SOM and in Chapter 3 to map a hierarchical (non-rectangular) map into a high dimensional data space. Unsupervised learning by prototype based approaches is still an very active field of research and, more recently, also low-dimensional embedding techniques, providing an unconstrained mapping to $2D$ or $3D$ have been proposed using prototype concepts see e.g. [26, 30].

Supervised models

In the field of prototype-based supervised learning large efforts have been made to obtain efficient models which are not only competitive to alternative strategies like SVM but also keep interpretability and sparsity. Two major cost functions dominate, namely the Generalized Learning Vector Quantizer (GLVQ) [95] and the Robust Soft Learning Vector Quantizer (RSLVQ) [106]. The latter can be interpreted as a probabilistic mixture model. For both approaches it has been shown that they are large margin optimizers with competitive performance to different alternative techniques [55].

Recent extensions of supervised prototype learners have focused widely on the topic

of metric adaptation [100]. Prototype approaches typically permit an easy replacement of the used dissimilarity measure by a general metric, with the typical constraint to be differentiable, for a recent summary see e.g. [14]. Basically the distance measure is replaced by a parametric distance. In general the Euclidean distance is used and replaced by a parametric form:

$$\begin{aligned} d(\mathbf{x}, \mathbf{w}_j) &= |\mathbf{x} - \mathbf{w}_j|^2 \\ d^{\lambda}(\mathbf{x}, \mathbf{w}_j) &= (\mathbf{x} - \mathbf{w}_j)^{\top} \Lambda (\mathbf{x} - \mathbf{w}_j) \end{aligned}$$

with the parameter matrix $\Lambda \in \mathbf{R}^{D \times D}$ and entries $\lambda_{i,j} \in \Lambda$, with appropriate constraints to provide a quadratic form. If Λ is a diagonal matrix with only ones at the diagonal we get back to the standard Euclidean distance. If we allow diagonal values $\lambda_{i,i} \in [0, 1]$ we get the weighted Euclidean distance, first used in prototype based learning in [21] and known as relevance learning. The λ-weights are typically also refereed to as a relevance profile, indicating the discrimination power of individual feature dimensions. Thereby a large $\lambda_{i,i}$ indicates a feature which is relevant for the learning task, low λ-values may indicate that a feature is unimportant, encoding noise, or is not necessary to improve the model performance, e.g. it may be redundant with respect to some other feature. More generic matrices Λ with different regularizations are discussed in [101] and typically referred to as matrix relevance learning. More recently also differently structured metrics have been considered typically based on domain specific considerations and are discussed in the Chapters 2, 3. The author and colleagues have also considered functional distance measures, which were found to be very promising in the life-science domain [97, 103, 126, 84]. Nowadays many different metrics have been analyzed for prototype based learning and even more flexibility is obtained by the similarity and dissimilarity learners as discussed in Chapter 4.

Semi-Supervised techniques in the field of prototype-based learning typically combine techniques from supervised or unsupervised learning and employ classical semi-supervised learning concepts, for recent work see e.g. [99]. Some methods discussed in Chapter 2 can be directly transferred to semi-supervised learning but have been initially defined for fully labeled data with fuzzy or unsafe label information.

Besides of GTM and SOM also some new techniques for prototype based data embedding, employing label information, have been published by the author and colleagues recently, like Limited Rank Matrix Learning [27] or a visualization technique by learning a projection function with auxiliary information [51].

Proximity learning and approximation strategies

In case of metric similarities, kernel machines have been widely used to define algorithms and models [108, 122] for this type of data. With the work of Platt [88] these methods are now also reasonably efficient for larger data sets and the concept of core-set machines [7, 119, 120] opened the door also for very large scale problems. Also prototype base learning methods can be extended to similarity and dissimilarity learners.

In general, prototypes are considered as vectors, as defined before, but for similarity or dissimilarity approaches see e.g. Chapter 4 the data may also be given only in form of a similarity matrix (e.g. a kernel), denoted as S or K or a dissimilarity matrix D with no explicit vector space and $S \in \mathbf{R}^{N \times N}$ and $D \in \mathbf{R}^{N \times N}$, with some moderate assumptions like symmetry. In these cases the prototypes are represented by means of linear combinations of points from the original data-space, or implicitly by considering similarities or dissimilarities like:

$$\mathbf{w}_j = \sum_l \alpha_{jl} \mathbf{x}_l \tag{1.2}$$

The coefficients $\alpha_{jl} \in [0, 1]$ are in general assumed to be normalized $\sum_l^M \alpha_{jl} = 1$. Most often it is assumed that the linear combination can be based on all data points so $M = N$ but this is not necessary and in particular for large sample sets some sampling strategies can be used. Additionally it is possible to define sparsity constraints on the underlying optimization problem to get most α_* close to zero as detailed in Chapter 4.

Prototypes in relational or kernel settings correspond to positions in pseudo-Euclidean or Hilbert space which are representative for the classes if measured according to the given similarity/dissimilarity measure. Thus, prototype inspection faces two problems: (i) the representation or embedding space is usually only implicit, (ii) it is not clear whether dimensions in this embedding carry any semantic information. Thus, albeit prototypes are represented as linear combinations of data also in the pseudo-Euclidean or kernel space setting, it is not clear whether these linear combinations correspond to a semantic meaning.

One approach which is taken in this context is to approximate a prototype by one or several exemplars, i.e. members of the data set, which are close by, also called k-approximation [52]. Thereby, the approximation can be improved if sparsity constraints for the prototypes are integrated during training [98].

This way, every prototype is represented by only a small number of exemplars which can be inspected like regular data. Again it is possible to visualize data and prototypes using some nonlinear dimensionality reduction technique. Naturally, both techniques, a representation of prototypes by few exemplars as well as a projection to low dimensions incorporate errors depending on the dimensionality of the pseudo-Euclidean or kernel space and its deviation from the Euclidean norm. Appropriate mapping approaches are not discussed in this work but can be found e.g. in [28].

For many data like text documents a vectorial representation is not available or complicated to obtain and often implicit representation by means of similarities or dissimilarities are used. Also many domain specific measures of relatedness of different object can lead to such data. Accordingly, these data are often given as large matrices of (dis-)similarity values.

Different strategies are necessary to enable prototype models for similarities and dissimilarities also at large scale. To keep learning tractable, the concept of the Nyström approximation is presented in Chapter 4. One strategy to obtain prototype models on (dis-)similarity data is to represent the prototypes implicitly by means of a linear combination of the original data points. This leads to a coefficient matrix which can be very dense, a phenomenon which, in a slightly different way, is also a problem for many kernel machines. Learning (dis-)similarity data by prototype approaches is addressed in the Chapter 4 and Nyström- as well as sparsity concepts are provided to derive interpretable, sparse models.

Organization

A variety of my current and former work is focused on domain specific extensions of prototype based learning which also often improves the interpretability of the model results. In this contribution, I summarized multiple of my articles, published at high-ranked journals, related to interpretable models with a special focus on prototype based learning strategies, addressing some of the priorly mentioned problems.
Following the identified problem fields, the individual contributions are grouped around three major topics:

- *Classification with uncertainty*,
- *Improved evaluation, interpretation and domain knowledge integration*

- *Large scale models*

Classification with uncertainty

The chapter 2 addresses the problem of *uncertainty* or fuzziness in classification problems. For many supervised data analysis tasks a label, given to a data-point, is subject of uncertainty and dedicated methods are desirable. The specific problem of uncertainty in the labels instead of the measured features, was addressed in different ways before using e.g. probabilistic models or fuzzy set concepts [19, 96] but not in the context of prototype based learning. Another methodology closely related to this topic is the field of semi-supervised learning (SSL) [135, 50]. The most recent SSL algorithms also consider the case of unsafe label information and the priorly mentioned conformal prediction is closely linked.

Also the interpretability of fuzzy models got some attention recently [3] but again few work has been done in the prototype field in this line. In the Chapter *Classification with uncertainty* different strategies are analyzed which have been the basis on further work by the author and colleagues about this topic as listed in the additional references. Especially techniques for unsupervised prototype based learning have been extended to incorporate label information but also supervised cost functions where adapted to met the new requirements and to incorporate so called fuzzy labels. Thereby the focus is given to vectorial data but the methods can be transferred to proximity data, as discussed in Chapter 4, in a straight forward manner. More recent work in this line was provided in [71] discussing alternative strategies to incorporate label information in unsupervised prototype approaches. The corresponding models discussed in Chapter 2 permit easy interpretation and can still be inspected by human experts.

Improved evaluation, interpretation and domain knowledge integration

The incorporation of domain knowledge, or constraints often helps to solve data modeling problems and is also useful to improve the interpretability of the algorithms and the models. As reflected by a multitude of recent publications [87, 11, 124, 93, 91, 3, 25, 115], appropriate models are of wide interest and have been approached by different researches and in various communities.

The chapter 3 is dedicated to papers and methods of this type. As already outline before interpretable models have to fulfill multiple properties, additionally interpretability can be achieved in the models in different ways and at different levels of abstraction. Accordingly there is no one single way, but multiple strategies, often specific to the underlying algorithm in the backbone can be considered.

The concept of conformal prediction (CP) [128] provides interesting strategies to extend known algorithms in different ways, often leading to improved interpretability. In this chapter different ways are shown how CP can be integrated and used in prototype based learning. This concept is especially relevant to add measures of confidence to predictive models without direct probabilistic outputs. The underlying non-conformity measure, the key parameter of this concept, permits easy integration of domain knowledge in the learning process as shown in Chapter 3. This strategies also permits the identification of novel objects or outliers which are in parts covered in one of the contributions see 3.2. However the topic of outlier or novelty detection has been addressed from different fields, with contribution from classical statistics [41] to dedicated one class classifiers [113, 112] in the field of machine learning.

Humans are used to simplify problems and favor representations which are simple and easy to communicate e.g. by visualizations [123]. These visualizations can be enhanced by additional information to communicate the model decision and to indicate the intended interpretation as provided by the underlying mathematical model. Prominent examples can be found in the field of visual analytics [72] but it is also often argued that the underlying models may lead to misleading conclusions and again the limitations of these approaches, e.g. explaining non-linear effects, are not sufficiently communicated.

An alternative view is to identify the intrinsic representation of a given dataset [78, 18]. Also low dimensional embeddings or manifold learning got much attention in the last time [117, 129, 132, 10] and also for prototype based learning different new strategies where provided recently [92, 26, 29, 30, 133].

A model taking domain knowledge about the expected intrinsic data structure into account is shown in a contribution about hierarchical prototype based learning 3.3. Another topic of integrated domain knowledge, is metric learning [15, 5] also addressed in multiple ways in this chapter. In the approach provided in 3.4 the prototypes are actually simulated entities based on a lot of domain knowledge as a result of an optimization process. Here, the prototypes are involved in a mixture model and do not directly quantize the given data space as in the methods before. Also, recent approaches in sparse coding [77, 114], non-negative matrix factorization [64] and the calculation of independent components [85, 93] show links to this problem.

Large scale models

The scalability of modern data analysis algorithm to large data sets is still often limited by the inherent complexity of these approaches. Often these algorithms scale by $O(N^2)$ to $O(N^3)$, or even worse, limiting not only the applicability but also the interpretability. This problem is especially prominent for non-linear methods, like kernel algorithms [108] or Gaussian process approaches [90].

An example is the Support Vector Machine, which in its original formulation does not scale to large data sets. Strategies like decomposition [86] have been effectively used and more recently geometrical properties of core-sets [32] were used to derive specific algorithms for this type of algorithms [119, 121] to improve the runtime and often also memory complexity with guarantee bounds. Further prominent approximation strategies are e.g. in the line of Nyström approximation [134] or more recently quad-tree concepts [9, 131]. Often these concepts are derived from other disciplines [34] and most of these approximation strategies make specific assumptions about the intrinsic dimension of the data, the number of non-vanishing eigenvalues, the distribution of the data or other properties of the data set. Also the memory complexity is often an issue and typical sparsity concepts are used to approach the problem. Sparsity of a model can be achieved in very different ways, by appropriate probabilistic models, including sparse priors [118, 33] by modifications of the underlying cost function, using e.g. lasso techniques [58, 110, 116, 73] or other sparsity measures and constraints [114]. Often, especially in the context of visualization not only the model generation, but also the out of sample extension can become very costly and very recent work provides approximated projection functions [51, 131]. If the analysis task can be effectively split into multiple independent sub-problems also parallelization schemes have been proposed [40].

While strategies to approach large scale problems are of wide interest for many paradigms, they did not got much attention in the field of prototype based learning. Most prototype technique are online approaches and standard strategies like sub-sampling have been often used. With the advent of many batch approaches in prototype based learning, and larger

datasets large scale analysis became an issue, with early work shown in [2]. As discussed before, the coefficient matrix used to represent the prototype model can become very large but also the proximity matrix, encoding the data, can get huge for larger data sets. In Chapter 4 some novel strategies are shown, based on Nyström approximation and sparse approximations with a focus on clustering and classification problems.

The work presented in this thesis has evolved over almost six years and was published at different high-ranked journals and conferences. These contributions cover a variety of different computational and mathematical aspects of prototype based learning algorithms.

The articles constitute a representative overview of my work and are accompanied by a number of contributions to international conferences, book chapters and additional journal papers. Many of the papers discuss not only theoretical extensions and proposals of algorithms but contain also applications of these methods in different domains. A major application domain is in the life science with data obtained from mass spectrometry, nuclear magnetic resonance spectroscopy, remote sensing, electrophysiological measurements or other types of (bio-)chemical measurements.

The methods are however not limited to these application fields but parametrizations are possible, such that also applications in the context of e.g. image processing are considered. The more theoretically proposals are supported by convergence and generalization analyses, extensive parameter studies and complexity analyses. I also discuss topics like topology preservation, statistical learning theory and generalization ability.

The following contributions have been selected because

- each article constitutes a major contribution to a specific area within the topic *Learning with interpretable models*
- the articles are non-redundant and cover different computational aspects or models; they contain work which is not contained in my PhD thesis
- at least 25% of the text of the following articles where I am a coauthor and which are based on joint work and discussions with colleagues is written by myself. I provide additional comments in the preface of each article to clearly state the contribution of the individual authors

Since the final layout of the articles is partially not yet available or not available in electronic form for the public, I compiled the following pages directly from the final versions accepted for publication, thereby substituting the layout of the journal by a uniform style, such that the following articles differ with respect to the layout from the published version or version to be published, respectively. No scientific content, formulation, or figure has been altered compared to the accepted versions.

Chapter 2

Classification with uncertainty

2.1 Fuzzy Classification by Fuzzy Labeled Neural Gas

The article *Fuzzy Classification by Fuzzy Labeled Neural Gas*, by T. Villmann, B. Hammer, F.**Schleif**, T. Geweniger and W. Herrmann was accepted by*Neural Networks* 19 (6-7), p. 772-779, in 2006. In the article a new semi-supervised learning algorithm was proposed. The theoretical derivation of the algorithm was done by T. Villmann, B. Hammer and myself. I implemented the algorithm and did the experiments for the simulated data. The clinical data and expertise was provided by W. Herrmann and with experiments using FLNG done by T. Geweniger. T. Villmann and I wrote the main parts of the article. The clinical discussion was written by T. Villmann and W. Herrmann. All authors discussed the general article. This article describes the basic concepts of the corresponding US patent $7,991,223B2$ (Villmann, Schleif, Hammer).

Additional publications using or based on this method I am co-author of include:

1. F.-M. **Schleif**, T. Villmann, B. Hammer, M. v.d. Werff, A. Deelder and R. Tollenaar, *Analysis of spectral data in clinical proteomics by use of learning vector quantizers*, In Studies in Computational Intelligence, Volume 151, 2008, Pages 141-167, ISBN: 978-354070776-9, 2008 (Content: The concept of Fuzzy Labeled Neural Gas is transfered to self-organized maps and used to process clinical mass spectrometry data)

2. T. Villmann, F.-M. **Schleif** and B. Hammer, *Prototype based fuzzy classification with local relevance for proteomics*, Neurocomputing 69 (16-18), Pages 2425-2428 (Content: A fuzzified, local version of the Soft Nearest Prototype Classifier is introduced and applied to the analysis of mass spectrometry data)

3. T. Villmann, F.-M. **Schleif**, M. v.d.Werff, A. Deelder, R. Tollenaar *Association learning in SOMs for fuzzy-classification*, In Proceedings of 6th International Conference on Machine Learning and Applications, ICMLA 2007 2007, Article number 4457292, Pages 581-586, ISBN: 0769530699;978-076953069-7 (Content: Short article introducing the basic concepts of the Fuzzy-Labeled-Self-Organizing-Map)

4. T. Villmann, U. Seiffert, F.-M. **Schleif**, C. Brüss, T. Geweniger, B. Hammer *Fuzzy labeled self-organizing map with label-adjusted prototypes*, 2nd IAPR Workshop on Artificial Neural Networks in Pattern Recognition, ANNPR 2006, Volume 4087 LNAI, 2006, Pages 46-56, ISBN: 3540379517;978-354037951-5 (Content: Fuzzy labeled self-organizing map are applied to plant biology data)

5. B. Hammer, A. Hasenfuss, F.-M. **Schleif**, T. Villmann 2nd IAPR Workshop on Artificial Neural Networks in Pattern Recognition, ANNPR 2006, Volume 4087 LNAI, 2006, Pages 33-45, ISBN: 3540379517;978-354037951-5 (Content: a semi-supervised variant of batch neural gas is proposed taking ideas of FLNG)

Fuzzy Classification by Fuzzy Labeled Neural Gas

Th. Villmann[1]*, B. Hammer[2], F. Schleif[3],
T. Geweniger[4], and W. Herrmann[5]
[1]University Leipzig, Clinic for Psychotherapy
[2]Clausthal Univ. of Tech., Dept. of Math. and CS
[3]Bruker Daltonik GmbH Leipzig and Univ. Leipzig, Dept. of CS
[4]University of Applied Science Mittweida
[5]Paracelsus-Hospital Zwickau

August 8, 2012

Abstract

We extend neural gas for supervised fuzzy classification. In this way we are able to learn crisp as well as fuzzy clustering, given labeled data. Based on the neural gas cost function, we propose three different ways to incorporate the additional class information into the learning algorithm. We demonstrate the effect on the location of the prototypes and the classification accuracy. Further, we show that relevance learning can be easily included.

Keywords: learning vector quantization, relevance learning, metric adaptation, classification

1 Introduction

Clustering is an important data processing task relevant for pattern recognition, sequence and image processing, data compression, etc. One appropriate tool is offered by prototype based vector quantization including effective concrete algorithms such as the Self-Organizing Map (SOM) and the Neural Gas network (NG) [11],[14]. These algorithms distribute the prototypes in a way that the data density is estimated by minimizing some description error aiming at *unsupervised* data clustering. Prototype based classification as a *supervised* vector quantization scheme is dedicated to distribute prototypes in a manner that data classes can be detected, which naturally is also influenced by the data density. Important approaches are the family LVQ [11] and the recent developments like Generalized LVQ (GLVQ) [18] or Supervised NG (SNG) [3]. Thereby, general parameterized distance measures can be applied and their parameters may also may be subject of the optimization. This paradigm is called relevance learning giving the respective algorithms GRLVQ and SRNG [4],[3].

One major assumption of these classification approaches is that both (training) data and prototype assignments to classes have to be crisp, i.e. a unique assignment of the data to the classes as well as for the prototypes is required. The latter restriction can be smoothed by

* *corresponding author, University Leipzig, Clinic for Psychotherapy, Karl-Tauchnitz-Str. 25, 04107 Leipzig, Germany, email: villmann@informatik.uni-leipzig.de*

a subsequent post-labeling of the prototypes after learning according to their responsibility to the training data yielding fuzzy assignments [20]. However, there do not exist supervised prototype based approaches to work with fuzzy labels in data during training so far, although they would be desirable. In real world applications for classification like in medicine, a clear (crisp) classification of training data may be difficult or impossible: Assignments of a patient to a certain disorder frequently can be done only in a probabilistic (fuzzy) manner. Hence, it is of great interest to have a classifier which is able to manage this type data.

In this contribution we provide modifications of the usual NG for solving fuzzy classification tasks. For this purpose we extend the cost function of the NG to incorporate the assessment of the fuzzy label accuracy. We obtain new learning schemes for the prototypes and additionally an adaptation rule for the update of the prototype fuzzy labels. We describe the effect of the learning schemes on the prototype locations and classification. Further we are able to integrate the relevance learning ideas for metric adaptation into this approach.

2 The neural gas network

Neural gas is an unsupervised prototype based vector quantization algorithm. It maps data vectors $\mathbf{v}$ from a (possibly high-dimensional) data manifold $\mathcal{D} \subseteq \mathbb{R}^d$ onto a set A of neurons i formally written as $\Psi_{\mathcal{D}\to A} : \mathcal{D} \to A$. Each neuron i is associated with a pointer $\mathbf{w}_i \in \mathbb{R}^d$ also called weight vector. All weight vectors establish the set $\mathbf{W} = \{\mathbf{w}_i\}_{i\in A}$. The mapping description is a winner take all rule, i.e. a stimulus vector $\mathbf{v} \in \mathcal{D}$ is mapped onto the neuron $s \in A$ the pointer $\mathbf{w}_s$ of which is closest to the actually presented stimulus vector $\mathbf{v}$ (winner),

$$\Psi_{\mathcal{D}\to A} : \mathbf{v} \mapsto s(\mathbf{v}) = \underset{i\in A}{\operatorname{argmin}}\, \xi(\mathbf{v}, \mathbf{w}_i). \tag{2.1}$$

whereby $\xi(\mathbf{v}, \mathbf{w})$ is usually the Euclidean norm $\xi(\mathbf{v}, \mathbf{w}) = \|\mathbf{v} - \mathbf{w}\| = (\mathbf{v} - \mathbf{w})^2$. Here we only suppose that it is a differentiable symmetric similarity measure.

During the adaptation process a sequence of data points $\mathbf{v} \in \mathcal{D}$ is presented to the map with respect to the data distribution $P(\mathcal{D})$. Each time the currently most proximate neuron s according to (2.1) is determined, and the pointer $\mathbf{w}_s$ as well as all pointers $\mathbf{w}_i$ of neurons in the neighborhood of $\mathbf{w}_s$ are shifted towards $\mathbf{v}$, according to

$$\triangle \mathbf{w}_i = -\epsilon h_\sigma(\mathbf{v}, \mathbf{W}, i) \frac{\partial \xi(\mathbf{v}, \mathbf{w}_i)}{\partial \mathbf{w}_i}. \tag{2.2}$$

The property of "being in the neighborhood of $\mathbf{w}_s$" is captured by the neighborhood function

$$h_\sigma(\mathbf{v}, \mathbf{W}, i) = \exp\left(-\frac{k_i(\mathbf{v}, \mathbf{W})}{\sigma}\right), \tag{2.3}$$

with the rank function

$$k_i(\mathbf{v}, \mathbf{W}) = \sum_j \theta\left(\xi(\mathbf{v}, \mathbf{w}_i) - \xi\left(\mathbf{v}, \mathbf{w}_j\right)\right) \tag{2.4}$$

counting the number of pointers $\mathbf{w}_j$ for which the relation $\|\mathbf{v} - \mathbf{w}_j\| < \|\mathbf{v} - \mathbf{w}_i\|$ is valid [14]. $\theta(x)$ is the Heaviside-function. We remark that the neighborhood function is evaluated in the input space. The adaptation rule for the weight vectors follows in average a potential dynamic according to the potential function [14]:

$$E_{NG} = \frac{1}{2C(\sigma)} \sum_j \int P(\mathbf{v})\, h_\sigma(\mathbf{v}, \mathbf{W}, j)\, \xi\left(\mathbf{v}, \mathbf{w}_j\right) d\mathbf{v} \tag{2.5}$$

with $C(\sigma)$ being a constant. It will be dropped in the following. It was shown in many applications that the NG shows a robust behavior together with a high precision of learning [5],[11],[15],[22],[23].

3 Fuzzy Labeled NG

We now switch from the unsupervised scheme to a supervised scenario, i.e. each data vector is now accompanied by a label. According to the aim of the paper, the label is fuzzy: for each class k we have the possibilistic assignment $x_k \in [0,1]$ collected in the label vector $\mathbf{x} = (x_1, \ldots, x_{N_c})$. N_c is the number of possible classes. Further, we introduce fuzzy labels for each prototype $\mathbf{w}_j$: $\mathbf{y}_j = \left(y_1^j, \ldots, y_{N_c}^j\right)$. Now, we adapt the original unsupervised NG so that it is able to learn the fuzzy labels of the prototypes according to a supervised learning scheme. Thereby, the behavior of the original NG should be integrated as much as possible to transfer the excellent learning properties. We denote this new algorithm Fuzzy Labeled Neural Gas (FLNG). To include the fuzzy label accuracy into the cost function of FLNG we add a term to the usual NG cost function, which judges the deviations of the prototype fuzzy labels from the fuzzy label of the data vectors:

$$E_{FLNG} = (1-\beta)\, E_{NG} + \beta E_{FL} \tag{3.1}$$

The factor β is a balance factor which could be under control or simply chosen as $\beta = 0.5$. Hence, we try a balancing between statistical properties of prototypes (nearest mean) and best classification accuracy, as it was also proposed in [21]. For precise definition of the new term E we have to differentiate between discrete and continuous data, which becomes clear during the derivation. We begin with the discrete case.

3.1 FLNG for discrete data

In the discrete case we have data $\mathbf{v}^k$ with labels $\mathbf{x}^k$. We define the additional term of the cost function as

$$E_{FL} = \frac{1}{2} \sum_j \sum_k h_\sigma \left(\mathbf{v}^k, \mathbf{W}, j\right) \left(\mathbf{x}^k - \mathbf{y}_j\right)^2 \tag{3.2}$$

To obtain the update rules for the weights and their labels, we take the derivative of E_{FLNG} with respect to $\mathbf{w}_i$ and $\mathbf{y}_i$. The latter one is simply obtained as

$$\begin{aligned} \frac{\partial E_{FLNG}}{\partial \mathbf{y}_i} &= \frac{\partial E_{FL}}{\partial \mathbf{y}_i} && (3.3) \\ &= -\sum_k h_\sigma \left(\mathbf{v}^k, \mathbf{W}, i\right) \left(\mathbf{x}^k - \mathbf{y}_i\right) && (3.4) \end{aligned}$$

which is a weighted average of all fuzzy labels of data.

For the weight vector update one takes the gradient $\frac{\partial E_{FLNG}}{\partial \mathbf{w}_i}$. The first term $\frac{\partial E_{NG}}{\partial \mathbf{w}_i}$ is known from usual NG, eq. (2.2). Considering the second term E_{FL} we get

$$\begin{aligned} \frac{\partial E_{FL}}{\partial \mathbf{w}_i} &= \frac{1}{2} \sum_j \sum_k \frac{\partial h_\sigma \left(\mathbf{v}^k, \mathbf{W}, j\right)}{\partial \mathbf{w}_i} \left(\mathbf{x}^k - \mathbf{y}_j\right)^2 && (3.5) \\ &= -\frac{1}{2\sigma} \sum_j \sum_k \frac{\partial k_j \left(\mathbf{v}^k, \mathbf{W}\right)}{\partial \mathbf{w}_i} h_\sigma \left(\mathbf{v}^k, \mathbf{W}, j\right) \left(\mathbf{x}^k - \mathbf{y}_j\right)^2 && (3.6) \end{aligned}$$

We introduce

$$\triangle\left(\mathbf{v},\mathbf{w}_i,\mathbf{w}_l\right)=\xi\left(\mathbf{v},\mathbf{w}_i\right)-\xi\left(\mathbf{v},\mathbf{w}_l\right) \tag{3.7}$$

and consider

$$\frac{\partial k_j\left(\mathbf{v}^k,\mathbf{W}\right)}{\partial\mathbf{w}_i}=\delta_{i,j}\cdot\sum_l\delta\left(\triangle\left(\mathbf{v}^k,\mathbf{w}_j,\mathbf{w}_l\right)\right)\frac{\partial\xi\left(\mathbf{v}^k,\mathbf{w}_j\right)}{\partial\mathbf{w}_i}-\delta\left(\triangle\left(\mathbf{v}^k,\mathbf{w}_j,\mathbf{w}_i\right)\right)\frac{\partial\xi\left(\mathbf{v}^k,\mathbf{w}_i\right)}{\partial\mathbf{w}_i} \tag{3.8}$$

with $\delta\left(x\right)$ being the Dirac-distribution and $\delta_{i,j}$ the Kronecker-symbol. So we obtain in (3.6)

$$\frac{\partial E_{FL}}{\partial\mathbf{w}_i} = -\frac{1}{2\sigma}\sum_k\left(\sum_l\delta\left(\triangle\left(\mathbf{v}^k,\mathbf{w}_i,\mathbf{w}_l\right)\right)\frac{\partial\xi\left(\mathbf{v}^k,\mathbf{w}_i\right)}{\partial\mathbf{w}_i}\right)h_\sigma\left(\mathbf{v}^k,\mathbf{W},i\right)\left(\mathbf{x}^k-\mathbf{y}_i\right)^2 \tag{3.9}$$

$$+\frac{1}{2\sigma}\sum_j\sum_k\left(\delta\left(\triangle\left(\mathbf{v}^k,\mathbf{w}_j,\mathbf{w}_i\right)\right)\frac{\partial\xi\left(\mathbf{v}^k,\mathbf{w}_i\right)}{\partial\mathbf{w}_i}\right)h_\sigma\left(\mathbf{v}^k,\mathbf{W},j\right)\left(\mathbf{x}^k-\mathbf{y}_j\right)^2 \tag{3.10}$$

which contributes only for vanishing $\triangle$-function, i.e. on the borders of the receptive fields of the neurons. However, in case of discrete data the probability for this is zero. Thus, the weight vector learning in the discrete scenario based on this cost function is (almost surely) independent of the label adaptation.

3.2 FLNG for continuous data

In case of continuous data the above argument is not valid: We cannot ignore the borders of the receptive fields. Therefore, it is impossible to treat the problem in the same way. As the consequence we redefine the term E_{FL} in (3.1) to avoid these difficulties. In the following we denote (continuous) data by $\mathbf{v}$ and its labels by $\mathbf{x}$.

3.2.1 Gaussian kernel

As the first method, we weight the label error by a Gaussian kernel depending on the distance between data point and considered prototype. Thus, we choose the second term E_{FL} as

$$E_{FL}=\frac{1}{2}\sum_j\int P\left(\mathbf{v}\right)g_\gamma\left(\mathbf{v},\mathbf{w}_j\right)\left(\mathbf{x}-\mathbf{y}_j\right)^2d\mathbf{v} \tag{3.11}$$

where $g_\gamma\left(\mathbf{v},\mathbf{w}_j\right)$ is a Gaussian kernel describing a neighborhood range in the data space using the distance measure $\xi\left(\mathbf{v},\mathbf{w}_j\right)$:

$$g_\gamma\left(\mathbf{v},\mathbf{w}_j\right)=\exp\left(-\frac{\xi\left(\mathbf{v},\mathbf{w}_j\right)}{2\gamma^2}\right) \tag{3.12}$$

Note that $g_\gamma\left(\mathbf{v},\mathbf{w}_j\right)$ depends on the prototype locations, such that E_{FL} is influenced by both $\mathbf{w}$ and $\mathbf{y}$. Investigating this cost function, again, the first term $\frac{\partial E_{NG}}{\partial\mathbf{w}_i}$ of the full gradient $\frac{\partial E_{FLNG}}{\partial\mathbf{w}_i}$ is known from usual NG. The new second term now contributes according to

$$\frac{\partial E_{FL}}{\partial\mathbf{w}_i} = \frac{1}{2}\sum_j\int P\left(\mathbf{v}\right)\frac{\partial g_\gamma\left(\mathbf{v},\mathbf{w}_j\right)}{\partial\mathbf{w}_i}\left(\mathbf{x}-\mathbf{y}_j\right)^2d\mathbf{v} \tag{3.13}$$

$$= -\frac{1}{4\gamma^2}\sum_j\int P\left(\mathbf{v}\right)g_\gamma\left(\mathbf{v},\mathbf{w}_j\right)\frac{\partial\xi\left(\mathbf{v},\mathbf{w}_j\right)}{\partial\mathbf{w}_i}\left(\mathbf{x}-\mathbf{y}_j\right)^2d\mathbf{v} \tag{3.14}$$

$$= -\frac{1}{4\gamma^2}\int P\left(\mathbf{v}\right)g_\gamma\left(\mathbf{v},\mathbf{w}_i\right)\frac{\partial\xi\left(\mathbf{v},\mathbf{w}_i\right)}{\partial\mathbf{w}_i}\left(\mathbf{x}-\mathbf{y}_i\right)^2d\mathbf{v} \tag{3.15}$$

which takes the accuracy of fuzzy labeling into account for the weight update. Both terms define the learning rule for the weights.

For the fuzzy label we simply obtain $\frac{\partial E_{FLNG}}{\partial \mathbf{y}_i} = \frac{\partial E_{FL}}{\partial \mathbf{y}_i}$, where

$$\frac{\partial E_{FL}}{\partial \mathbf{y}_i} = -\int P(\mathbf{v})\, g_\gamma(\mathbf{v}, \mathbf{w}_i)\, (\mathbf{x} - \mathbf{y}_i)\, d\mathbf{v} \tag{3.16}$$

which is, in fact, a weighted average of the data fuzzy labels of those data belonging to the receptive field of the associated prototypes. However, in comparison to usual NG the receptive fields are different because of the modified learning rule for the prototypes and their resulting different locations. The resulting learning rule is

$$\triangle \mathbf{y}_i = \epsilon_l \beta g_\gamma(\mathbf{v}, \mathbf{w}_i)\, (\mathbf{x} - \mathbf{y}_i) \tag{3.17}$$

3.2.2 Approximation of the rank function

As a second approach, we approximate the original neighborhood function h_σ. In (2.4) we replace the Heaviside-function by a sigmoid function

$$\zeta(x) = \frac{1}{1 + \exp\left(-\frac{x}{2\tau^2}\right)} \tag{3.18}$$

and obtain an approximation of the rank:

$$\tilde{k}_j(\mathbf{v}, \mathbf{W}) = \sum_l \zeta(\triangle(\mathbf{v}, \mathbf{w}_j, \mathbf{w}_l)) \tag{3.19}$$

using the $\triangle$-notation (3.7). Then the additional term of the cost function is defined as

$$\tilde{E}_{FL} = \frac{1}{2} \sum_j \int P(\mathbf{v})\, \tilde{h}_\sigma(\mathbf{v}, \mathbf{W}, j)\, \left(\mathbf{x} - \mathbf{y}_j\right)^2 d\mathbf{v} \tag{3.20}$$

with $\tilde{h}_\sigma(\mathbf{v}, \mathbf{W}, j) = \exp\left(-\frac{\tilde{k}_i(\mathbf{v}, \mathbf{W})}{\sigma}\right)$. To obtain the update rules we take the derivative of E_{FLNG} with respect to $\mathbf{w}_i$ and $\mathbf{y}_i$. The latter one is simply obtained as

$$\begin{aligned} \frac{\partial E_{FLNG}}{\partial \mathbf{y}_i} &= \frac{\partial \tilde{E}_{FL}}{\partial \mathbf{y}_i} & (3.21) \\ &= -\int P(\mathbf{v})\, \tilde{h}_\sigma(\mathbf{v}, \mathbf{W}, i)\, (\mathbf{x} - \mathbf{y}_i)\, d\mathbf{v} & (3.22) \end{aligned}$$

which is a weighted average of all fuzzy labels of the data.

For the weight vector update one takes the gradient $\frac{\partial E_{FLNG}}{\partial \mathbf{w}_i}$. The first term $\frac{\partial E_{NG}}{\partial \mathbf{w}_i}$ is known from usual NG, eq. (2.2). Considering the second term $\tilde{E}_{FL}$ we get

$$\begin{aligned} \frac{\partial \tilde{E}_{FL}}{\partial \mathbf{w}_i} &= \frac{1}{2} \sum_j \int P(\mathbf{v})\, \frac{\partial \tilde{h}_\sigma(\mathbf{v}, \mathbf{W}, j)}{\partial \mathbf{w}_i} \left(\mathbf{x} - \mathbf{y}_j\right)^2 d\mathbf{v} & (3.23) \\ &= -\frac{1}{2\sigma} \sum_j \int P(\mathbf{v})\, \frac{\partial \tilde{k}_j(\mathbf{v}, \mathbf{W})}{\partial \mathbf{w}_i} \tilde{h}_\sigma(\mathbf{v}, \mathbf{W}, j) \left(\mathbf{x} - \mathbf{y}_j\right)^2 d\mathbf{v} & (3.24) \end{aligned}$$

We derive

$$\frac{\partial \tilde{k}_j(\mathbf{v}, \mathbf{W})}{\partial \mathbf{w}_i} = \delta_{i,j} \cdot \left(\sum_l \zeta'(\triangle(\mathbf{v}, \mathbf{w}_i, \mathbf{w}_l)) \frac{\partial \xi(\mathbf{v}, \mathbf{w}_j)}{\partial \mathbf{w}_i} \right) - \zeta'(\triangle(\mathbf{v}, \mathbf{w}_j, \mathbf{w}_i)) \frac{\partial \xi(\mathbf{v}, \mathbf{w}_i)}{\partial \mathbf{w}_i} \tag{3.25}$$

with $\zeta'(x) = \frac{1}{2\tau^2}\zeta(x)(1-\zeta(x))$. So we obtain in (3.24)

$$\frac{\partial \tilde{E}_{FL}}{\partial \mathbf{w}_i} = -\frac{1}{2\sigma}\int P(\mathbf{v})\left(\sum_l \zeta'(\triangle(\mathbf{v},\mathbf{w}_i,\mathbf{w}_l))\frac{\partial \xi(\mathbf{v},\mathbf{w}_i)}{\partial \mathbf{w}_i}\right)\tilde{h}_\sigma(\mathbf{v},\mathbf{W},i)(\mathbf{x}-\mathbf{y}_i)^2 d\mathbf{v} \quad (3.26)$$

$$+\frac{1}{2\sigma}\sum_j \int P(\mathbf{v})\left(\zeta'(\triangle(\mathbf{v},\mathbf{w}_j,\mathbf{w}_i))\frac{\partial \xi(\mathbf{v},\mathbf{w}_i)}{\partial \mathbf{w}_i}\right)\tilde{h}_\sigma(\mathbf{v},\mathbf{W},j)\left(\mathbf{x}-\mathbf{y}_j\right)^2 d\mathbf{v} \quad (3.27)$$

Hence, the full update becomes

$$\triangle \mathbf{w}_i = -\left((1-\beta)\epsilon h_\sigma(\mathbf{v},\mathbf{W},i) - \frac{\beta\epsilon'}{\sigma}\tilde{h}_\sigma(\mathbf{v},\mathbf{W},i)(\mathbf{x}-\mathbf{y}_i)^2\sum_l \zeta'(\triangle(\mathbf{v},\mathbf{w}_i,\mathbf{w}_l))\right) \quad (3.28)$$

$$\cdot\frac{\partial \xi(\mathbf{v},\mathbf{w}_i)}{\partial \mathbf{w}_i} - \frac{\beta\epsilon'}{\sigma}\frac{\partial \xi(\mathbf{v},\mathbf{w}_i)}{\partial \mathbf{w}_i}\sum_j \tilde{h}_\sigma(\mathbf{v},\mathbf{W},j)\cdot\zeta'(\triangle(\mathbf{v},\mathbf{w}_j,\mathbf{w}_i))\cdot\left(\mathbf{x}-\mathbf{y}_j\right)^2 \quad (3.29)$$

The respective label update rule is obtained in complete analogy to (3.17) as

$$\triangle \mathbf{y}_i = \epsilon_l \beta \tilde{h}_\sigma(\mathbf{v},\mathbf{W},i)(\mathbf{x}-\mathbf{y}_i) \quad (3.30)$$

4 Relevance Learning in FLNG

In the theoretical derivation of the algorithm we have used a general distance measure, which can, in principle, be chosen arbitrarily. Now we consider the case of a parametrized, quadratic distance measure $\xi_\lambda(\mathbf{w}_i,\mathbf{w}_s)$ with parameters $\lambda = (\lambda_1,\ldots,\lambda_m)$. It has recently been demonstrated for both, supervised and unsupervised prototype based learning that an adaptation of the metric during training can greatly increase the accuracy without decreasing the usually excellent generalization ability [3],[2],[10]. Because of the mathematical derivation of FLNG by means of a cost function, the principle of learning metrics can be easily transferred to our approach. Here we demonstrate this fact by deriving the learning rules for the metric parameters λ. For this purpose we investigate the derivative

$$\frac{\partial E_{FLNG}}{\partial \lambda_k} = (1-\beta)\frac{\partial E_{NG}}{\partial \lambda_k} + \beta\frac{\partial E_{FL}}{\partial \lambda_k} \quad (4.1)$$

of the cost function. First we consider the continuous cases: The first term $\frac{\partial E_{NG}}{\partial \lambda_k}$ gives

$$\frac{\partial E_{NG}}{\partial \lambda_k} = \frac{1}{2C(\sigma)}\begin{pmatrix} \sum_j \int P(\mathbf{v})\, h_\sigma(\mathbf{v},\mathbf{W},j)\frac{\partial \xi_\lambda(\mathbf{v},\mathbf{w}_j)}{\partial \lambda_k}d\mathbf{v} \\ +\sum_j \int P(\mathbf{v})\,\xi_\lambda\left(\mathbf{v},\mathbf{w}_j\right)\frac{\partial h_\sigma(\mathbf{v},\mathbf{W},j)}{\partial \lambda_k}d\mathbf{v} \end{pmatrix} \quad (4.2)$$

with $\frac{\partial h_\sigma(\mathbf{v},\mathbf{W},j)}{\partial \lambda_k} = -\frac{h_\sigma(\mathbf{v},\mathbf{W},j)}{\sigma}\cdot\frac{\partial k_j(\mathbf{v},\mathbf{W})}{\partial \lambda_k}$. We take into account that the definition (2.4) of $k_j(\mathbf{v},\mathbf{W})$ with the derivative of the Heaviside-function $\theta(x)$ is the delta distribution $\delta(x)$. In this way we get

$$\frac{\partial k_j(\mathbf{v},\mathbf{W})}{\partial \lambda_k} = \sum_l \delta(\triangle_\lambda(\mathbf{v},\mathbf{w}_j,\mathbf{w}_l))\cdot\frac{\partial \triangle_\lambda(\mathbf{v},\mathbf{w}_j,\mathbf{w}_l)}{\partial \lambda_k} \quad (4.3)$$

with $\triangle_\lambda(\mathbf{v},\mathbf{w}_j,\mathbf{w}_l) = \xi_\lambda\left(\mathbf{v},\mathbf{w}_j\right) - \xi_\lambda(\mathbf{v},\mathbf{w}_l)$ using the notation (3.7). Hence, the second term in (4.2) vanishes because δ is symmetric and non-vanishing only for $\xi_\lambda\left(\mathbf{v},\mathbf{w}_j\right) =$

$\xi_\lambda(\mathbf{v}, \mathbf{w}_l)$. Thus

$$\frac{\partial E_{NG}}{\partial \lambda_k} = \frac{1}{2C(\sigma)} \sum_j \int P(\mathbf{v})\, h_\sigma(\mathbf{v}, \mathbf{W}, j)\, \frac{\partial \xi_\lambda\left(\mathbf{v}, \mathbf{w}_j\right)}{\partial \lambda_k} d\mathbf{v} \tag{4.4}$$

In the discrete case we simply replace the integration over the input data by the respective summation in (4.2).

We now pay attention to the second summand $\frac{\partial E_{FL}}{\partial \lambda_k}$. For the discrete case, we can apply the same arguments as above. Thus we get (almost surely)

$$\triangle \lambda_k = -\epsilon_\lambda (1-\beta) \sum_j \frac{\partial \xi_\lambda\left(\mathbf{v}^l, \mathbf{w}_j\right)}{\partial \lambda_k} h_\sigma\left(\mathbf{v}^l, \mathbf{W}, j\right) \tag{4.5}$$

For the choice of E_{FL} according to the kernel approach (3.11) we have

$$\begin{aligned} \frac{\partial E_{FL}}{\partial \lambda_k} &= \frac{1}{2} \sum_j \int P(\mathbf{v})\, \frac{\partial g_\gamma\left(\mathbf{v}, \mathbf{w}_j\right)}{\partial \lambda_k} \left(\mathbf{x} - \mathbf{y}_j\right)^2 d\mathbf{v} && (4.6) \\ &= -\frac{1}{4\gamma^2} \sum_j \int P(\mathbf{v})\, g_\gamma\left(\mathbf{v}, \mathbf{w}_j\right) \frac{\partial \xi\left(\mathbf{v}, \mathbf{w}_j\right)}{\partial \lambda_k} \left(\mathbf{x} - \mathbf{y}_j\right)^2 d\mathbf{v} && (4.7) \end{aligned}$$

Putting all together we obtain for the relevance adaptation of the distance parameter in the first continuous case:

$$\frac{\partial E_{FLNG}}{\partial \lambda_k} = \sum_j \int P(\mathbf{v})\, \frac{\partial \xi_\lambda\left(\mathbf{v}, \mathbf{w}_j\right)}{\partial \lambda_k} \left(\frac{(1-\beta)}{2C(\sigma)} h_\sigma(\mathbf{v}, \mathbf{W}, j) - \frac{\beta}{4\gamma^2} g_\gamma\left(\mathbf{v}, \mathbf{w}_j\right) \left(\mathbf{x} - \mathbf{y}_j\right)^2 \right) d\mathbf{v} \tag{4.8}$$

The second continuous case with the sigmoid approximation (3.19) gives

$$\frac{\partial \tilde{E}_{FL}}{\partial \lambda_k} = -\frac{1}{2\sigma} \sum_j \int P(\mathbf{v})\, \tilde{h}_\sigma(\mathbf{v}, \mathbf{W}, j)\, \frac{\partial \tilde{k}_j(\mathbf{v}, \mathbf{W})}{\partial \lambda_k} \left(\mathbf{x} - \mathbf{y}_j\right)^2 d\mathbf{v} \tag{4.9}$$

with

$$\frac{\partial \tilde{k}_j(\mathbf{v}, \mathbf{W})}{\partial \lambda_k} = \sum_l \zeta'\left(\triangle_\lambda(\mathbf{v}, \mathbf{w}_j, \mathbf{w}_l)\right) \frac{\partial \triangle_\lambda(\mathbf{v}, \mathbf{w}_j, \mathbf{w}_l)}{\partial \lambda_k},$$

which together with (4.1) and (4.4) gives the learning rule.

5 Experiments and Applications

In the following we give some experimental results. The data sets are an artificial one of overlapping Gaussian distributions whereas the second one is a medical application.

5.1 Artificial data set of overlapping Gaussians

First we apply the FLNG to an artificial data set of two overlapping Gaussian distributions with the same prior probability. Thereby, we used in the first experiments the usual Euclidean metric as distance measure $\xi_\lambda(\mathbf{v}, \mathbf{w}) = \xi(\mathbf{v}, \mathbf{w}) = \|\mathbf{v} - \mathbf{w}\|$. The classification results of the different FLNG versions in comparison to an usual post-labeled NG using 10 prototypes for balancing parameter $\beta = 0.5$ are depicted in Fig.1.

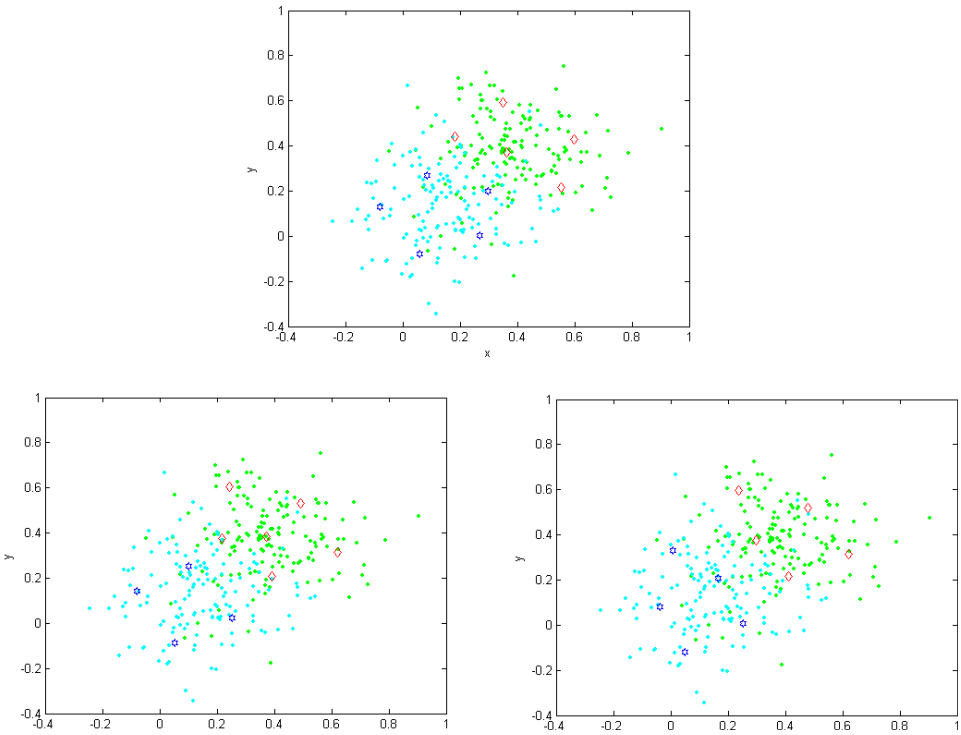


Figure 1: Comparison of the prototype distributions for the different FLNG approaches with $\beta = 0.5$. The influence of the E_{FL} is obviously. The prototypes symbols are according to the maximum component of their fuzzy labels: discrete FLNG (upper center), Gaussian FLNG (bottom left), sigmoid variant (bottom right).

β	NG (post)	discr. FLNG	Gaussian FLNG	sigmoid FLNG
0.3	82.0%	82.0%	82.2%	84.1%
0.5			83.7%	84.3%
0.7			85.3%	85.1%
0.9			82.0%	82.1%

Table 1: Classification accuracy for the artificial data set of overlapping Gaussians obtained by the several approaches for different β balancing parameter values.

One can clearly observe the influence of the label oriented part of the cost function by the additional prototype learning term $\frac{\partial E_{FL}}{\partial \mathbf{w}_i}$. As expected for this simple data set, the accuracy is only slightly improved by the proposed methods, see Tab.1. However, the distributions of the prototypes differ significantly: Thereby, the discrete variant yields similar results compared to post labeled NG, which can be expected from the learning rules, because the labels do not influence the prototype updates for the discrete version. Similarly, the results of the two continuous variants do not differ much from each other, which is due to the fact that the two data classes are unimodal. However, the continuous approaches place more prototypes nearby the class border. Thus, the class labels clearly influence the prototype location for these versions.

Obviously, β controls the influence of the label learning. Therefore we varied this parameter to demonstrate the effects. The respective prototype distribution are depicted exemplary for the Gaussian variant of FLNG in Fig.2.

One can observe the varying strength of influence with respect to the balancing parameter β, looking at the prototype distributions. The effect is also emphasized by the obtained classification accuracy, given in Tab.1. In particular, for high β-values the accuracy is decreased which can be addressed to the weak force for weight vector update according to the usual NG.

5.2 Classification of electrophysiological impairment profiles in case of Wilson's disease

In a more challenging application we consider the classification of patients suffering from Wilson's disease and volunteers according to their electrophysiological abilities. *Wilson's disease* is a rare autosomal-recessive disorder of copper metabolism. Patients suffering from Wilson's disease especially show disturbances in liver function and basal ganglia which lead to hepatic and extra-pyramidal motoric symptoms [1],[17]. The symptoms usually occur in an age range from 5 to 40 years with clinical heterogeneity and different severity [13],[16]. One distinguishes the neurological and the non-neurological type of disease depending on the severity of clinical symptoms. Thereby, the more disturbed type is the neurological case. Besides these clinical symptoms, Wilson's disease patients also exhibit subclinical disorders of other central nervous pathways which may be observed as latency prolongations in electrophysiological tests [8]. Yet, the severity of nervous pathway disorder also depends on the severity of Wilson's disease. In particular, electrophysiological impairments have increased severity for the neurological state of illness.

In our consideration we collected electrophysiological data from a standardized test set generating a so-called electrophysiological profile containing the test result of 7 different tests covering the typical features (EAEP, MSEP, TSEP, T-VEP, MEP, EEG, heart frequency variability) [8]. These profiles are taken as 7-dimensional data vectors. Overall the data set

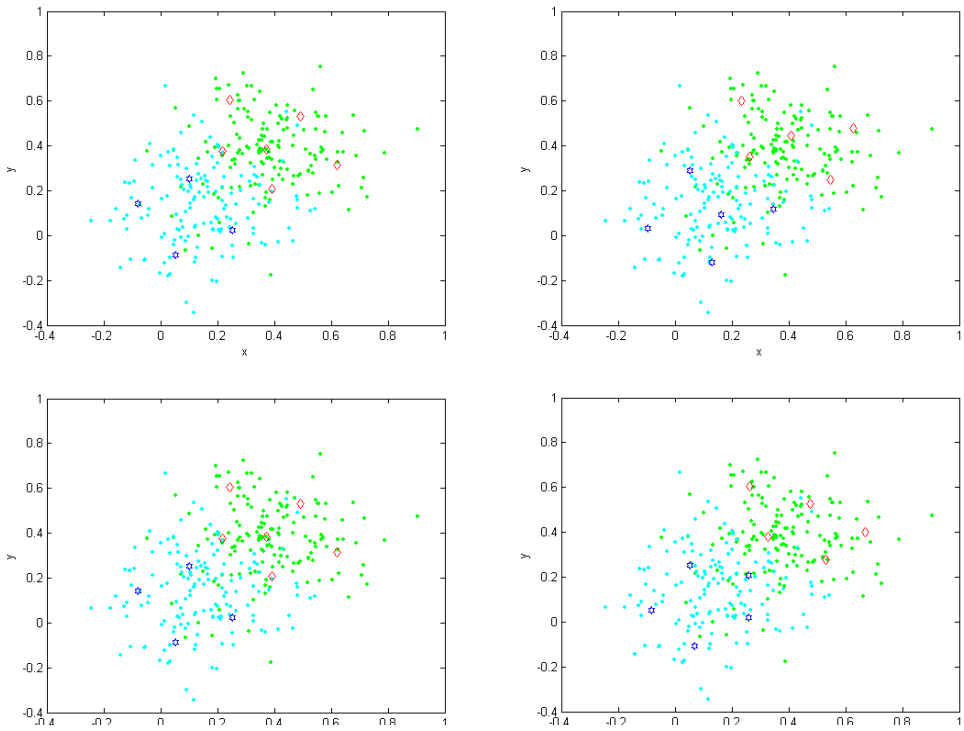


Figure 2: Comparison of the prototype distributions for the Gaussian variant of FLNG for different values β: discrete FLNG (upper-left), $\beta = 0.3$ (upper right), $\beta = 0.5$ (bottom left), $\beta = 0.9$ (bottom right).

	discr. FLNG	Gaussian FLNG	Gaussian FLNG w. relevance
train	81.8%	83.6%	85.3%
test	82.0%	83.3%	84.9%

Table 2: Classification accuracy for the Wilson data set obtained by the several approaches. The balancing parametr was $\beta = 0.5$.

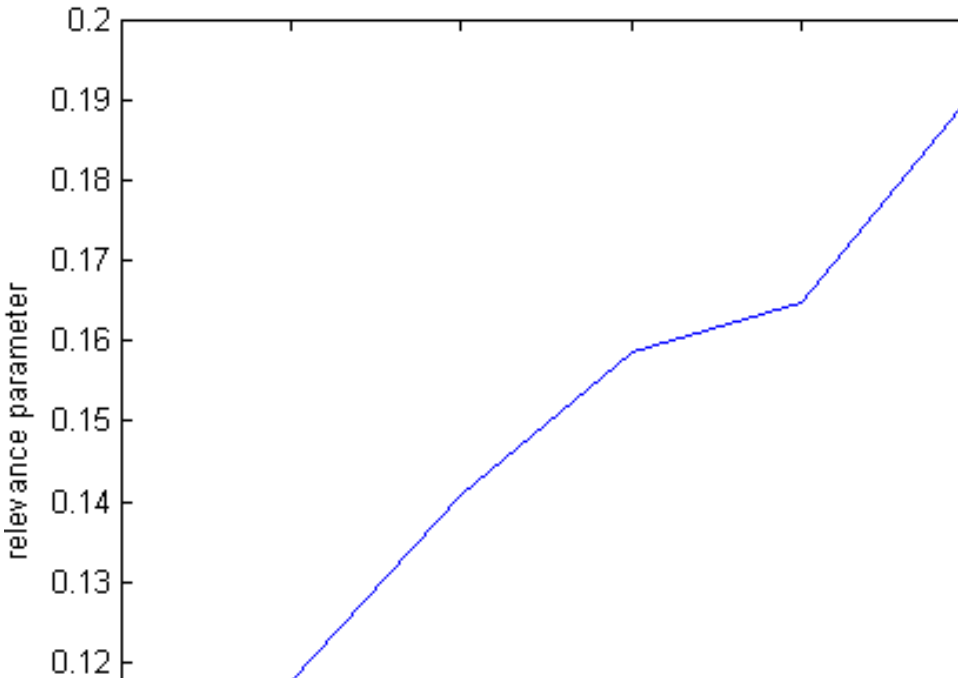


Figure 3: Comparison of the prototype distributions for the Gaussian variant of FLNG for different values β: discrete FLNG (upper-left), $\beta = 0.3$ (upper right), $\beta = 0.5$ (bottom left), $\beta = 0.9$ (bottom right).

consists of 37 patient data vectors of several severity stages and 24 volunteers data, whereby the patients group contains 8 non-neurological cases which should have no electrophysiological disturbances. From these data set we take 27 and 15 for training, respectively. Testing was performed using all data. Beside the classification of the electrophysiological profiles it is also of medical interest, which tests are most significantly separating electrophysiologically disturbed persons from the other. This is a typical problem for relevance learning in fuzzy classification using the scaled Euclidean distance $\xi_\lambda(\mathbf{v}, \mathbf{w}) = \sum_i \lambda_i (\mathbf{v} - \mathbf{w})^2$, $\lambda_i > 0$, $\sum_i \lambda_i = 1$.

We applied the discrete FLNG as well as the Gaussian variant using 6 prototypes. Additionally we included relevance learning. The balance parameter was chosen as $\beta = 0.75$. The results are depicted in Tab. 2. From relevance learning a weighting of the input dimensions respective tests is obtained. From this we can conclude that the EEG-test is most important for class decision, whereas EAEP, MSEP and heart rate frequency variability are less significant, see Fig.3.

The last result is in agreement with other clinical findings [7]. Thus, the determined

relevance profile reflects the characteristic of the data set well.

6 Discussion and Conclusion

We extended the usual unsupervised NG to a supervised fuzzy classification approach by means of an extension of the cost function. In this way we are able to give risk estimations of the classification accuracy. This is of particular interest e.g. in domains such as medical applications since, on the one hand data might come with fuzzy labeling; on the other hand, a judgment of the classification security is highly desirably. As demonstrated, there are different ways to model fuzzy labeling, ranging from a simple post labeling to cost functions where the labeling influences the location of the prototypes. We proposed three approaches based on a gradient descent of an extended NG cost function, explicitly including the class information of data. Thereby, the neighborhood cooperativeness of prototype learning is integrated into the label adaptation. Yet, the range of influence should be in agreement with the neighborhood influence of the usual NG. Hence, the parameters γ in (3.12) should be chosen such that the influence of the Gaussian kernel covers the same range as the neighborhood cooperativeness in $\frac{\partial E_{NG}}{\partial \mathbf{w}_i}$. For the sigmoid FLNG the neighborhood range of the label learning is influenced by the choice of the smoothness parameter ζ of the sigmoid function (3.18). A good choice should be ζ equal to the average of the pairwise distances of the data.

Comparing FLNG with the usual LVQ the following observation can be made: the new additional term $\frac{\partial E_{FL}}{\partial \mathbf{w}_i}$ for prototype update only contributes significantly if the label difference $\left(\mathbf{x} - \mathbf{y}_j\right)$ of an input $(\mathbf{v}, \mathbf{x})$ is high but the distance to the prototype $\xi_\lambda\left(\mathbf{v}, \mathbf{w}_j\right)$ is small. This can be interpreted that the term $\frac{\partial E_{FL}}{\partial \mathbf{w}_i}$ pushes the prototype away in this case, as it is known from LVQ for the best matching but wrong classifying prototype adaptation.

Experiments for artificial and real world data demonstrate the effect of these learning rules on the classification accuracy and location of prototypes. They show that the new approach is able to classify with high accuracy, reflecting the data structure well. Relevance learning can improve the result and gives further informations.

Obviously, the FLNG approach can be transferred to other vector quantization schemes, too, if a cost function for the method exists. Thus, usual SOM is not extendible to fuzzy labeling in this way, but its variant introduced by Heskes [9] would be another proper framework, which would be an alternative to the SOM based on auxiliary spaces introduced by Kaski [10],[19]. Last but not least information theoretic vector quantization methods as described in [6],[12] also would be an interesting field of fuzzy labeling extension.

References

[1] J.A. Cuthbert. Wilson's disease. update of a systemic disorder with protean manifestations. *Gastroenterology Clinics of North America*, 27(3):655–681, 1998.

[2] B. Hammer, M. Strickert, and Th. Villmann. On the generalization ability of GRLVQ networks. *Neural Processing Letters*, 21(2):109–120, 2005.

[3] B. Hammer, M. Strickert, and Th. Villmann. Supervised neural gas with general similarity measure. *Neural Processing Letters*, 21(1):21–44, 2005.

[4] B. Hammer and Th. Villmann. Generalized relevance learning vector quantization. *Neural Networks*, 15(8-9):1059–1068, 2002.

[5] B. Hammer and Th. Villmann. Mathematical aspects of neural networks. In M. Verleysen, editor, *Proc. Of European Symposium on Artificial Neural Networks (ESANN'2003)*, pages 59–72, Brussels, Belgium, 2003. d-side.

[6] A. Hegde, D. Erdogmus, T. Lehn-Schioler, Y.N. Rao, and J.C. Principe. Vector quantization by density matching in the minimum Kullback-Leibler-divergence sense. In *Proc. of the International Joint Conference on Artificial Neural Networks (IJCNN) - Budapest*, pages 105–109, IEEE Press, 2004.

[7] W. Hermann, P. Gnther, A. Wagner, and T. Villmann. Klassifikation des Morbus Wilson auf der Basis neurophysiologischer Parameter. *Der Nervenarzt*, 76:733–739, 2005.

[8] W. Hermann, Th. Villmann, and A. Wagner. Elektrophysiologisches Schädigungsprofil von Patienten mit einem Morbus Wilson'. *Der Nervenarzt*, 74(10):881–887, 2003.

[9] T. Heskes. Energy functions for self-organizing maps. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 303–316. Elsevier, Amsterdam, 1999.

[10] Samuel Kaski, Janne Sinkkonen, and Jaakko Peltonen. Bankruptcy analysis with self-organizing maps in learning metrics. *IEEE Transactions on Neural Networks*, 12:936–947, 2001.

[11] Teuvo Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 1995. (Second Extended Edition 1997).

[12] T. Lehn-Schiler, A. Hegde, D. Erdogmus, and J.C. Principe. Vector quantization using information theoretic concepts. *Natural Computing*, 4(1):39–51, 2005.

[13] J. Lössner, H. Bachmann, R. Siegemund, H.-J. Kühn, and K. Günther. Wilsonsche Erkrankung in der DDR: Rückblick und Ausblick - eine Bilanz. *Psychiatrie, Neurologie und medizinische Psychologie*, 10:585–600, 1990.

[14] Thomas M. Martinetz, Stanislav G. Berkovich, and Klaus J. Schulten. 'Neural-gas' network for vector quantization and its application to time-series prediction. *IEEE Trans. on Neural Networks*, 4(4):558–569, 1993.

[15] M. Oja, S.l Kaski, and T. Kohonen. Bibliography of self-organizing map (SOM) papers: 1998-2001 addendum. *Neural Computing Surveys*, 3:1–156, 2003.

[16] E. A. Roberts and D.W. Cox. Wilson disease. *Bailliere's Clinical Gastroenterology*, 12(2):237–256, 1998.

[17] T. Saito. Presenting symptoms and natural history of wilson disease. *European Journal of Pediatrican*, 146:261–265, 1987.

[18] A. Sato and K. Yamada. Generalized learning vector quantization. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8. Proceedings of the 1995 Conference*, pages 423–9. MIT Press, Cambridge, MA, USA, 1996.

[19] Janne Sinkkonen and Samuel Kaski. Clustering based on conditional distributions in an auxiliary space. *Neural Computation*, 14:217–239, 2002.

[20] G. Van de Wouwer, P. Scheunders, D. Van Dyck, M. De Bodt, F. Wuyts, and P.H. Van de Heyning. Wavelet-FILVQ classifier for speech analysis. In *Proc. of the Int. Conf. Pattern Recognition*, pages p. 214–218, Vienna, 1996. IEEE Press.

[21] C.J. Veenman and M.J.T. Reinders. The nearest subclass classifier: A compromise between the nearest mean and nearest neighbor classifier. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9), 2005.

[22] T. Villmann and J.-C. Claussen. Magnification control in self-organizing maps and neural gas. *Neural Computation*, 18(2):446–469, February 2006.

[23] Th. Villmann, U. Seiffert, and A. Wismüller. Theory and applications of neural maps. In M. Verleysen, editor, *European Symposium on Artificial Neural Networks 2004*, pages 25–38. d-side publications, 2004.

2.2 Analysis of Spectral Data in Clinical Proteomics by use of Learning Vector Quantizers

In the article *Classification of Mass-Spectrometric Data in Clinical Proteomics Using Learning Vector Quantization Methods* a novel semi-supervised topographic mapping was proposed by T. Villmann, F.-M.**Schleif**, M. Kostrzewa, A. Walch and B. Hammer. It appeared in 2008 in the Journal Briefings in Bioinformatics 9 (2), p. 129-143. The first two authors derived and implemented the algorithm. The tissue samples and imaging measurements where provided by A. Walch, who also provided the clinical discussion. The bacterial data and corresponding protein spectra where provided by M. Kostrzewa. B. Hammer and T. Villmann supervised the project. All authors discussed the paper.

Classification of Mass-Spectrometric Data in Clinical Proteomics Using Learning Vector Quantization Methods

T. Villmann, F.-M. Schleif, M. Kostrzewa,
A. Walch and B. Hammer

August 8, 2012

Abstract

In the present contribution we present two recently developed classification algorithms for analysis of mass-spectrometric data - the supervised neural gas and the fuzzy labeled self-organizing map. The algorithms are inherently regularizing, which is recommended, for these spectral data because of its high dimensionality and the sparseness for specific problems. The algorithms are both prototype based such that the principle of characteristic representants is realized. This leads to an easy interpretation of the generated classifcation model. Further, the fuzzy labeled self-organizing map, is able to process uncertainty in data, and classification results can be obtained as fuzzy decisions. Moreover, this fuzzy classifcation together with the property of topographic mapping offers the possibility of class similarity detection, which can be used for class visualization. We demonstrate the power of both methods for two exemplary examples: the classification of bacteria (listeria types) and neoplastic and non-neoplastic cell populations in breast cancer tissue sections.

1 Introduction

Exploration and analysis of mass spectrometric data in the field of clinical proteomics have become one of the key problems in computational proteomics. Thereby, the complexity of the mass spectrometric data is one difficult problem. Frequently, the data are given as huge-dimensional functional vectors with several thousands dimensions according to the resolution on the mass axis. Further, usually the number of samples is limited to a few data sets due to clinical restrictions. From am athematical point of view, the data space to be explored is sparsely filled. Thereby, the spectra may be overlaid by noise such that the contained signal is difficult to extract. A further problem arises for data analysis methods as consequence of the high dimensionality: the data can always be separated more or less independent of the separation criteria [32]. Thus, any method is confronted with the problem of the detection of the underlying regularities. Usually, this problem is overcome by cross-validation. Yet, the certainty of such an evaluation is diminished here, because of the humble number of data. Therefore, advanced methods for data analysis in mass spectrometry are required to be regularizing inherently, robust and to be able to deal with high-dimensional, sparse and noisy data.

In the following we will restrict us to classification problems. Thereby, we will concentrate to the following aspects: How we can achieve a good classification accuracy and how we can visualize classification results in an adequate manner. The latter problem is related to the

problem of class similarity detection. Moreover, each classification result depends on the underlying similarity/dissimilarity measure for data.

Classification in traditional statistics is frequently realized by Fisher's discriminant analysis (FDA) or linear/quadratic discriminant analysis (LDA/QDA) [24]. FDA optimizes the inter-intra-class correlation ratio by weighting the data dimensions to obtain a good separation plane, i.e. it is based on a weighted Euclidean distance for data similarity. LDA/QDA tries to optimize the Bayes error of the classification by utilization of the (class dependent, QDA) covariance, i.e. the Mahalanobis distance between data is used inherently [10]. These classical statistic approaches are more and more supplemented by machine learning tools, which provide adaptive and robust methods for pattern recognition in complex data [1],[26],[27]. Thereby, machine learning algorithms comprise approaches like artificial neural networks (ANNs), evolutionary algorithms (EAs), decision trees (DTs), clustering, and other [4].

Beside the pure classification accuracy of a generated classification model, its interpretability plays an important role. Standard methods use (linear) principal component analysis (PCA) and Fisher's discriminant analysis or classical hierarchical clustering [8],[9]. Additionally, advanced preprocessing procedures, like denoising using wavelets or 'intelligent' peak picking heuristics including problem specific expert knowledge, are applied to improve the accuracy. Here, the flexibility of machine learning methods offers new ways which may result in better results [23]. For example, multilayer perceptron neural networks (MLPs) as universal function approximators offer, on the one hand side, greatest flexibility in learning and adaptation to achieve good classification results [5]. On the other hand, however, their decision scheme is more or less a 'black box', because all the information for the decision is distributed over the whole network. In contrast, prototype based classifiers realize the principle of '*characteristic representatives'* for data subsets or decision regions between them . Thus, the interpretation becomes easy. Examples for such tools are Support Vector Machines (SVM) [28], Kohonen's Learning Vector Quantization (LVQ), Self-Organizing Maps (SOMs) [18] and respective variants. New developments include the utilization of non-standard metrics (functional norms, scaled Euclidean metric) and task-dependent automatic metric adaptation (feature selection), fuzzy classification, and similarity based visualization of data. These properties offers new possibilities for analysis also of mass spectrometric data.

In the present paper we will give insights to two recently developed prototype based classifiers which fulfill the above requirements. The Supervised Neural Gas (SRNG) and the Fuzzy labeled SOM (FLSOM) are robust prototype based neural classifiers, which are inherently regularizing by neighborhood cooperativeness between prototypes and which are easy to interpret. Moreover, as we will explain FLSOM is able to detect class similarities and offers the possibility of fuzzy classification. Both algorithms share the flexibility of utilization of arbitrary data metrics, which may be adapted during the training process as well in dependence on the classification task to be learned [15].

The article is structured as follows. First, we shortly review both methods, classification based on LVQ and FLSOM, pointing out their different properties and abilities. Thereby, we will emphasize the ability of the usage of general, task adequate, similarity measures in both methods. Further, we will highlight the class similarity detection probability of the semi-supervised FLSOM, which can be used for adequate class visualization or clinical interpretation. The theoretic part is followed by two clinical example investigations. The first one is an investigation of mass spectrometric bacteria data to find an adequate classification. In the second application a classification of neoplastic and non-neoplatic cell populations in histological sections of breast cancer tissue is considered. For both applications we demonstrate the advanced abilities of the methods. Concluding remarks complete

the paper.

2 Prototype based classifiers

Usually, spectrometric data in proteomic analysis are given as vectors $\mathbf{v} \in V \subseteq \mathbb{R}^D$. D is the data dimension which may be huge in this field. It depends on the spectral range and sampling resolution. Because the data represent spectra, or more general functions, they are called some times *functional data*. We remark that for functional data, the sequence of data dimensions is not independent.

In our consideration we assume, that there exist an underlying (unknown) data probability density P in V. Further, we assume for the training data that to each data vector $\mathbf{v}$ a unique class label $\mathbf{c}(\mathbf{v})$ exist. Prototype based classifiers distributes prototypes $\mathbf{w_r} \in \mathbb{R}^D$, $\mathbf{r} \in A$, as representations for classes in the data space V, whereby A is a given index set. The prototypes should represent class distributions in the data space and borders between different classes. For this purpose, each prototype has a class label $\mathbf{y_r}$.

Several approaches exist: the well-known LVQ family introduced by KOHONEN tries to minimize the Bayes classification error, but the adaptation dynamic is only a heuristic Hebbian like and does not perform a gradient descent on the misclassification error [18]. SVMs are based on structural risk optimization using a separation margin maximization approach [7]. Both methods are very powerful. In particular, SVMs frequently show superior results [28]. However, if new data become available for training a complete new learning has to be applied for SVM. Both methods have in common that they are not able to handle uncertainty in classification for training data (fuzzy class memberships). Further, the obtained classification model is crisp.

We now review two recently developed classification schemes, which both are inherently regularizing to address the above mentioned problem of noisy and sparse data in huge-dimensional data spaces. The first one is a generalization of Kohonen's LVQ scheme providing a gradient descent on a cost function. The second one extend the unsupervised SOM, such that a semi-supervised fuzzy classifier is obtained with excellent visualization abilities and the feature of class similarity detection. Both methods share the ability to proceed arbitrary (differentiable) may be parametrized data similarity measures, which itself can be in parallel subject of optimization with respect to the classification task.

2.1 Classification by Supervised Neural Gas

As mentioned above, LVQ does not minimize the classification error by gradient descent prototype adaptation. Therefore SATO&YAMADA introduced a cost function based on a classification function μ such that the respective gradient descent is similar to the heuristic LVQ learning scheme preserving the Hebbian characteristic [22]. For a given data point $\mathbf{v}$ with class label $\mathbf{c}(\mathbf{v})$ the two best matching prototypes with respect to the data metric d, usually the quadratic Euclidian, are determined: $\mathbf{w_{r^+}}$ has minimum distance $d^+ = d(\mathbf{v}, \mathbf{w_{r^+}})$ and the class labels are identically: $\mathbf{y_{r^+}} = \mathbf{c}(\mathbf{v})$. The other best prototype $\mathbf{w_{r^-}}$ has has minimum distance $d^- = d(\mathbf{v}, \mathbf{w_{r^-}})$ but the class labels are different: $\mathbf{y_{r^-}} = \mathbf{c}(\mathbf{v})$. Then the classification function $\mu(\mathbf{v})$ is defined as

$$\mu(\mathbf{v}) = \frac{d^+ - d^-}{d^+ + d^-} \tag{1}$$

The value $d^+ - d^-$ yields the hypothesis margin of the classifier [6]. Then the *generalized* LVQ (GLVQ) is derived as gradient descent of the cost function

$$C_{GLVQ} = \sum_{\mathbf{v}} f\left(\mu\left(\mathbf{v}\right)\right) \tag{2}$$

with respect to the prototypes. f is the sigmoid function

$$f\left(x\right) = \frac{1}{1 + \exp\left(-x\right)} \tag{3}$$

In one learning step for a given data point, both $\mathbf{w}_{\mathbf{r}^+}$ and $\mathbf{w}_{\mathbf{r}^-}$ are adapted in parallel. Taking the derivative yields the updates

$$\triangle \mathbf{w}_{\mathbf{r}^+} = \epsilon^+ \cdot \mathrm{sgd}'_{\mu(\mathbf{v})} \cdot \xi^+ \cdot \frac{\partial\left[d\left(\mathbf{v}, \mathbf{w}_{\mathbf{r}^+}\right)\right]}{\partial \mathbf{w}_{\mathbf{r}^+}}$$

and

$$\triangle \mathbf{w}_{\mathbf{r}^-} = -\epsilon^- \cdot \mathrm{sgd}'_{\mu(\mathbf{v})} \cdot \xi^- \cdot \frac{\partial\left[d\left(\mathbf{v}, \mathbf{w}_{\mathbf{r}^-}\right)\right]}{\partial \mathbf{w}_{\mathbf{r}^-}}$$

where ϵ^+ and $\epsilon^- \in (0, 1)$ are the learning rates. The logistic function $f\left(x\right)$ is evaluated at position $\mu(\mathbf{v})$, and we get

$$\xi^+ = \frac{2 \cdot d^-}{(d^+ + d^-)^2}$$

and

$$\xi^- = \frac{2 \cdot d^+}{(d^+ + d^-)^2}.$$

Yet, so far no inherently regularization is involved in the classification model. This feature can be included by combination of GLVQ with an unsupervised neural prototype vector quantizer - the neural gas (NG) [21]. NG realizes a Hebbian learning of prototypes together with regularization by neighborhood cooperativeness between prototypes. The level of cooperativeness is determined in dependence on the similarity of the prototypes to a given data vector: Let $\mathbf{W}$ be the set of prototypes and $L\left(\mathbf{v}, \mathbf{W}\right)$ be the ordered list of prototype indices such that for each pair $\mathbf{r}_i, \mathbf{r}_k \in L$ with $i < k$ the relation $d\left(\mathbf{v}, \mathbf{w}_{\mathbf{r}_i}\right) \leq d\left(\mathbf{v}, \mathbf{w}_{\mathbf{r}_k}\right)$ holds. Then the position $i\left(\mathbf{r}\right)$ denotes the rank of the competition of the prototypes to be the best matching for $\mathbf{v}$. The degree of cooperativeness is defined by

$$h_\sigma^{NG}\left(\mathbf{r}, \mathbf{v}, \mathbf{W}\right) = \exp\left(\frac{-i\left(\mathbf{r}\right)}{2\sigma^2}\right)$$

with neighborhood range σ determining the regularization strength. High values σ lead to strong regularization whereas low values relax this restriction [21].

Including this regularization scheme into GLVQ the supervised neural gas (SNG) is obtained [13]. For this purpose, we modify the GLVQ cost function (2) to

$$C_{SNG} = \frac{1}{C\left(\sigma, N_{\mathbf{W}_{\mathbf{c}(\mathbf{v})}}\right)} \sum_{\mathbf{v}} h_\sigma^{NG}\left(\mathbf{r}, \mathbf{v}, \mathbf{W}_{\mathbf{c}(\mathbf{v})}\right) \cdot f\left(\mu^{\mathbf{r}}\left(\mathbf{v}\right)\right)$$

whereby, $\mathbf{W}_{\mathbf{c}(\mathbf{v})}$ is the subset of all prototypes $\mathbf{w}_\mathbf{r}$ the class labels $\mathbf{y}_\mathbf{r}$ of which are equal to the class label $\mathbf{c}\left(\mathbf{v}\right)$ of the data point. $C\left(\sigma, N_{\mathbf{W}_{\mathbf{c}(\mathbf{v})}}\right)$ is a constant depending on the cardinality $N_{\mathbf{W}_{\mathbf{c}(\mathbf{v})}}$ of the subset $\mathbf{W}_{\mathbf{c}(\mathbf{v})}$ and the regularization level σ. Derivation of this cost

function leads to a similar adaptation scheme as for GLVQ. However, now a neighborhood cooperativeness is included between all prototypes of the correct class:

The update formulas for the prototypes can be obtained taking the derivative. For each $\mathbf{v}$, all prototypes $\mathbf{w_r} \in \mathbf{W_{c(v)}}$ are adapted by

$$\triangle \mathbf{w_r} = \epsilon^+ \cdot \frac{\mathrm{sgd}'|_{\mu^{\mathbf{r}}(\mathbf{v})} \cdot \xi_{\mathbf{r}}^+ \cdot h_\sigma^{NG}\left(\mathbf{r}, \mathbf{v}, \mathbf{W_{c(v)}}\right)}{C\left(\sigma, N_{\mathbf{W_{c(v)}}}\right)} \cdot \frac{\partial\left[d\left(\mathbf{v}, \mathbf{w_r}\right)\right]}{\partial \mathbf{w_r}}$$

and the closest wrong prototype is adapted by

$$\triangle \mathbf{w_{r^-}} = -\epsilon^- \cdot \sum_{\mathbf{w_r} \in \mathbf{W_{c(v)}}} \frac{\mathrm{sgd}'|_{\mu^{\mathbf{r}}(\mathbf{v})} \cdot \xi_{\mathbf{r}}^- \cdot h_\sigma^{NG}\left(\mathbf{r}, \mathbf{v}, \mathbf{W_{c(v)}}\right)}{C\left(\sigma, N_{\mathbf{W_{c(v)}}}\right)} \cdot \frac{\partial\left[d\left(\mathbf{v}, \mathbf{w_{r^-}}\right)\right]}{\partial \mathbf{w_{r^-}}}$$

whereby ϵ^+ and $\epsilon^- \in (0, 1)$ are learning rates and the logistic function is evaluated at position

$$\mu^{\mathbf{r}}(\mathbf{v}) = \frac{d_{\mathbf{r}} - d_{\mathbf{r}^-}}{d_{\mathbf{r}} + d_{\mathbf{r}^-}}\,.$$

The terms $\xi_{\mathbf{r}}^+$ and $\xi_{\mathbf{r}}^-$ are obtained as

$$\xi_{\mathbf{r}}^+ = \frac{2 \cdot d_{\mathbf{r}^-}}{(d_{\mathbf{r}} + d_{\mathbf{r}^-})^2}$$

and

$$\xi_{\mathbf{r}}^- = \frac{2 \cdot d_{\mathbf{r}}}{(d_{\mathbf{r}} + d_{\mathbf{r}^-})^2}\,.$$

Note that the updates of GLVQ are recovered for vanishing regularization $\sigma \to 0$. We remark that SNG also optimizes the hypothesis margin because the cost function contains the term $d_{\mathbf{r}} - d_{\mathbf{r}^-}$.

The final classification of unknown data points $\mathbf{v}$ is then realized by a winner take all mapping for both GLVQ and SNG:

$$\mathbf{v} \mapsto c(\mathbf{v}) = \mathbf{y_r} \text{ such that } d(\mathbf{v}, \mathbf{w_r}) \text{ is minimum.} \tag{4}$$

It was been demonstrated that SNG/GLVQ achieve excellent classification results [13], [35].

2.2 Semi-supervised fuzzy classification by fuzzy labeled SOM and class similarity detection

2.2.1 The fuzzy labeled SOM - FLSOM

We now turn to the more general task of fuzzy classification and classification visualization. For this purpose we assume that the number N_c of potential classes is known in advance. Then the class label $\mathbf{c}\left(\mathbf{v}\right)$ is taken as a class membership vector $\mathbf{c}\left(\mathbf{v}\right) \in \mathbb{R}^{N_c}$ the elements $c_i\left(\mathbf{v}\right) \in [0, 1]$ of which describe the fuzzy degree of class membership of the data vector $\mathbf{v}$ and sum up to $\sum_i c_i = 1$. Analogously, the prototype labels are taken as vectors $\mathbf{y_r} \in \mathbb{R}^{N_c}$. In the following we will extend the unsupervised SOM model to deal with classification tasks.

SOMs are powerful models for unsupervised vector quantization [18]. In SOMs the index set A is a regular grid, usually a rectangular or hexagonal two-dimensional lattice.

The indices $\mathbf{r}$ of the prototypes now indicate a location in the grid and, therefore, a natural metric $\|\cdot\|_A$ between them is induced. The mapping is like in SNG/GLVQ again a winner take all rule, which reads in the HESKES' variant of SOMs as

$$\mathbf{v} \mapsto s\left(\mathbf{v}\right) = \underset{\mathbf{r}\in A}{\operatorname{argmin}} \sum_{\mathbf{r}'\in A} h_\sigma(\mathbf{r},\mathbf{r}') \cdot d\left(\mathbf{v}, \mathbf{w}_{\mathbf{r}'}\right) \tag{5}$$

with

$$h_\sigma(\mathbf{r},\mathbf{r}') = \exp\left(-\frac{\|\mathbf{r}-\mathbf{r}'\|_A}{2\sigma^2}\right) \tag{6}$$

as neighborhood function [16]. It performs a topographic mapping of data under certain conditions [34], i.e. similar data points are mapped onto the same or onto neighbored grid locations[1]. The degree of topology preservation can be estimated by the *topographic product* TP [2]. TP-values nearby zero indicate adequate topographic mapping. For optimum results the lattice size and dimension can be dynamically adapted during learning [3]. This growing SOM (GSOM) generates a *non-linear* PCA of the data [33].

The learning in the Heskes-SOM follows a gradient descent on a cost function:

$$E_{\mathrm{SOM}} = \frac{1}{2C(\sigma)} \int P(\mathbf{v}) \sum_{\mathbf{r}} \delta_{\mathbf{r}}^{s(\mathbf{v})} \sum_{\mathbf{r}'} h_\sigma(\mathbf{r},\mathbf{r}') \xi(\ \mathbf{v}, \mathbf{w}_{\mathbf{r}'}) d\mathbf{v} \tag{7}$$

where $C\left(\sigma\right)$ is a constant, which we will drop in the following, and $\delta_{\mathbf{r}}^{\mathbf{r}'}$ is the usual Kronecker symbol checking the identity of $\mathbf{r}$ and $\mathbf{r}'$. All prototypes are adapted according to

$$\triangle \mathbf{w}_{\mathbf{r}} = -\epsilon h_\sigma\left(\mathbf{r}, s(\mathbf{v})\right) \frac{\partial \xi\left(\mathbf{v}, \mathbf{w}_{\mathbf{r}}\right)}{\partial \mathbf{w}_{\mathbf{r}}} \tag{8}$$

with learning rate $\epsilon > 0$.

Now we extend the cost function of the SOM as defined in (7) to a cost function for semi-supervised fuzzy classification by

$$E_{\mathrm{FLSOM}} = \left(1-\beta\right) E_{\mathrm{SOM}} + \beta E_{\mathrm{FL}} \tag{9}$$

where E_{FL} measures the classification accuracy . The factor $\beta \in [0,1]$ is a factor balancing unsupervised and supervised learning. One can simply choose $\beta = 0.5$, for example. We choose

$$E_{\mathrm{FL}} = \frac{1}{2} \int P\left(\mathbf{v}\right) \sum_{\mathbf{r}} g_\gamma\left(\mathbf{v}, \mathbf{w}_{\mathbf{r}}\right) \left(\mathbf{x} - \mathbf{y}_{\mathbf{r}}\right)^2 d\mathbf{v} \tag{10}$$

where $g_\gamma\left(\mathbf{v}, \mathbf{w}_{\mathbf{r}}\right)$ is a Gaussian kernel describing a neighborhood range in the data space:

$$g_\gamma\left(\mathbf{v}, \mathbf{w}_{\mathbf{r}}\right) = \exp\left(-\frac{d\left(\mathbf{v}, \mathbf{w}_{\mathbf{r}}\right)}{2\gamma^2}\right). \tag{11}$$

This choice is based on the assumption that data points close to the prototype determine the corresponding label if the underlying classification is sufficiently smooth. Note that $g_\gamma\left(\mathbf{v}, \mathbf{w}_{\mathbf{r}}\right)$ depends on the prototype locations, such that E_{FL} is influenced by both $\mathbf{w}_{\mathbf{r}}$ and $\mathbf{y}_{\mathbf{r}}$. Hence, prototype adaptation is now influenced by the classification task via the labels:

$$\frac{\partial E_{\mathrm{FLSOM}}}{\partial \mathbf{w}_{\mathbf{r}}} = \frac{\partial E_{\mathrm{SOM}}}{\partial \mathbf{w}_{\mathbf{r}}} + \frac{\partial E_{\mathrm{FL}}}{\partial \mathbf{w}_{\mathbf{r}}} \tag{12}$$

[1]For a detailed discussion of topographic mapping and more general lattice structures we refer to [3],[34].

which yields

$$\begin{aligned} \triangle \mathbf{w_r} &= -\epsilon(1-\beta)\cdot h_\sigma\left(\mathbf{r}, s(\mathbf{v})\right)\frac{\partial d\left(\mathbf{v},\mathbf{w_r}\right)}{\partial \mathbf{w_r}} \\ &\quad +\epsilon\beta\frac{1}{4\gamma^2}\cdot g_\gamma\left(\mathbf{v},\mathbf{w_r}\right)\frac{\partial d\left(\mathbf{v},\mathbf{w_r}\right)}{\partial \mathbf{w_r}}\left(\mathbf{x}-\mathbf{y_r}\right)^2 . \end{aligned} \tag{13}$$

The label adaptation is only influenced by the second part E_{FL}. The derivative $\frac{\partial E_{\mathrm{FL}}}{\partial \mathbf{y_r}}$ yields

$$\triangle \mathbf{y_r} = \epsilon_l\beta\cdot g_\gamma\left(\mathbf{v},\mathbf{w_r}\right)\left(\mathbf{x}-\mathbf{y_r}\right) \tag{14}$$

with learning rate $\epsilon_l > 0$. This label learning performs to a weighted average of the data fuzzy labels of those data close to the associated prototypes.

Classification of unknown data is again obtained by the mapping rule (4), but now giving a fuzzy class membership vector response. Usually, the classification accuracy of FLSOM is slightly less then the accuracy of a pure (good) classifier, because the balancing parameter β cannot be set to the unit due to numerical stability reasons [37]. Hence, a remaining unsupervised ammount of data information may lead to reduced accuracy. However, this disadvantage is compensated by the feature of inherent class similarity detection and the visualization abilities of FLSOM [38].

2.2.2 Class visualization and class similarity detection

As mentioned above, unsupervised SOMs generate a topographic mapping from the data space onto the prototype grid A under specific conditions. If the classes are consistently determined with respect to the varying data in a classification problem, one can expect for the semi-supervised topographic FLSOM that the class labels $\mathbf{y_r}$ become ordered within the distribution over the grid structure of the lattice A. In this case the topological order of the prototypes should be transferred to the topological order of prototype labels, such that we have a smooth change between the fuzzy class label vectors within the neighbored of the considered grid locations. This is the consequence of the following fact: the neighborhood function $h_\sigma\left(\mathbf{r},\mathbf{s}\right)$ of the usual SOM learning (8) forces the topological ordering of the prototypes. In FLSOM, this ordering is further influenced by the weighted classification error

$$ce\left(\mathbf{v},\mathbf{r}\right) = g_\gamma\left(\mathbf{v},\mathbf{w_r}\right)\left(\mathbf{x}-\mathbf{y_r}\right)^2 , \tag{15}$$

which contains the data space neighborhood $g_\gamma\left(\mathbf{v},\mathbf{w_r}\right)$, eq. (11). Hence, the prototype ordering contains information of both data density and class distribution, whereby for high balancing value β the latter term becomes dominant. Otherwise, the data space neighborhood $g_\gamma\left(\mathbf{v},\mathbf{w_r}\right)$ also triggers the label learning (14), which, of course, also dependents on the underlying learned prototype distribution and ordering. Thus, a consistent ordering of the labels is obtained in FLSOM [36].

As a consequence, the evaluation of the similarities between the prototype label vectors yields suggestions for the *similarity of classes*, i.e. similar classes are represented by prototypes in a local spatial area of the FLSOM lattice A. In case of overlapping class distributions the topographic processing leads to prototypes with unclear decision (labels), located between prototypes with clear vote. Further, if classes are not distinguish-able, there will exist prototypes responsive to those data, which have class label vectors containing approximately the same degree of fuzzy class membership for the respective classes.

The fuzzy class membership vectors allow an easy visualization of the classification using their similarity property. For this purpose, all label vectors $\mathbf{y_r}$, $\mathbf{r} \in A$ are embedded into a

color space preserving their similarities. This can be realized by multi-dimensional scaling (MDS), for example. Doing so, similar classes are coded by similar colors, which may be used for *class visualization* [38].

2.3 Classification task dependent metric adaptation

The dissimilarity measure $d(\mathbf{v}, \mathbf{w_r})$ for the data space V is usually chosen as squared Euclidean metric in GLVQ, SNG and FLSOM. Thus the derivative $\frac{\partial d(\mathbf{v},\mathbf{w})}{\partial \mathbf{w}}$ simply becomes $-2(\mathbf{v}-\mathbf{w})$. Depending on the classification task, this choice could be not optimum. Therefore, more appropriate (differentiable) similarity measures can be plugged into these algorithms instead, reflecting the nature of data or structure of classification. For example, Lee&Verleysen proposed a *functional metric* derived from the general *Minkowski-metric* for functional data paying attention to the spatial correlation between the components of functional vectors [19]. Other example, frequently used in biological and biochemical problems, are the Pearson correlation [29] and the Tanimoto kernel [30]. Due to the general formulation of the methods above, these metrics can easily plugged into the algorithms.

Yet, more flexibility can be obtained if $d(\mathbf{v}, \mathbf{w_r})$ is a parametrized similarity measure. Then, the respective parameters may be also subject of optimization according to the given classification task [14],[13].

Generally, we consider a parametrized distance measure $d^{\lambda}(\mathbf{v}, \mathbf{w})$ with a parameter vector $\lambda = (\lambda_1, \ldots, \lambda_M)$ with $\lambda_i \geq 0$ and normalization $\sum_{i=1}^{M} \lambda_i = 1$. Then, a classification task depending parameter optimization is achieved again by a gradient descent of the above cost functions but here with respect to these metric parameters.

One important example of a parametrized metric is the *scaled* squared Euclidean metric

$$d^{\lambda}(\mathbf{v}, \mathbf{w}) = \sum_i \lambda_i (v_i - w_i)^2 \tag{16}$$

(with $\lambda_i \geq 0$ and $\sum_i \lambda_i = 1$). The derivative $\frac{\partial d^{\lambda}(\mathbf{v},\mathbf{w})}{\partial \mathbf{w}}$ becomes $= -2 \cdot \mathbf{\Lambda} \cdot (\mathbf{v} - \mathbf{w})$ with $\mathbf{\Lambda}$ is a diagonal matrix and its i-th diagonal entry is λ_i and $\frac{\partial d^{\lambda}(\mathbf{v},\mathbf{w})}{\partial \lambda_i} = (v_i - w_i)^2$.

The parameter optimization of the *scaled* squared Euclidean metric allows a useful interpretation. The parameter λ_i weight the dimensions of the data space. Hence, optimization of these parameters in dependence on the classification problem leads to a ranking of the input dimensions according to their classification decision relevance. Therefore, metric parameter adaptation of the scaled Euclidean metric is called *relevance learning* [14]. This weighting is structurally similar to the weighting in FDA. In case of zero-valued λ_i relevance learning can also be seen as feature selection. The vector λ is called *relevance* profile. It can be used for advanced data investigation as it is shown in the applications.

3 Application of GRLVQ and FLSOM for clinical data sets

In this section we demonstrate the application of both, SNG and FLSOM, to classification of two spectrometric data sets in bioinformatics. The problems are characteristic for bioinformatic tasks and therefore exemplary:

1. identification of bacteria
2. breast cancer tissue slice classification

For both problems, the data metric for FLSOM was chosen as the squared scaled Euclidian metric (16).

3.1 Description of data and preprocessing

The data for both problems are based on mass-spectrometric profiles measured by linear MALDI-TOF MS devices from Bruker Daltonik, Bremen, Germany.

3.1.1 The bacteria data set

The bacteria samples are obtained from extracts from listeria cell cultures (original culture stems from the German Resource Center for Biological Material – DSMZ). The extracts, covered by a HCCA matrix, have been applied onto the MSP 96 target ground steel [20]. Profiling spectra were generated on a linear *Autoflex* MALDI-TOF MS. Details can be found in [17].

Listeria is a bacterial genus containing six species. This species consist of *listeria monocytogenes*, *listeria innocua*, *listeria ivanovii*, *listeria seeligeri*, *listeria welshimeri* and *listeria grayi*. Listeria occur very common in nature environments and are also present in water, plants, food and the bowel of humans. The identification of listeria is therefore an important problem in biology. Listeria are known to be the bacteria responsible for listeriosis, a rare but lethal food-borne infection that has a devastating mortality rate of 25% [31]. Listeria, also has a particularly high occurrence rate in newborns because of its ability to infect the fetus by penetrating the endothelial layer of the placenta [31]. Thereby Listeria monocytogenes is considered to be pathogenic for humans. Listeria in food are relatively rare but due to the increasing industrial production of food with many processing steps, the risk of a listerial contamination is increasing, which rises the needs for improved product safety and quality control. The diagnosis of listeria at an early stage is important for therapeutic approaches on humans. The expression of a infection caused by listeria may delay upto 8 weeks. To identify whether a listeria infection is present, the blood or matter is taken from the patient and a cultivation is tried. This, however, fails in part and, hence, the disease can not be diagnosed in time.

In the available data set all six listeria types are present. Thereby, for the listeria grayi a subgroup of *listeria grayi murrei* can be identified and for listeria ivanovii a distinction into the subgroups *listeria ivanovii ssp ivanovii* and *listeria ivanovii ssp londoniensis* can be made. Thus, the data set consists of 109 profile spectra in 8 classes with at least 6 samples for each class. The spectral range is between 2kDa and 20kDa. The obtained spectra have been smoothed, baseline corrected and, aligned following the standardized preprocessing tool BIOTYPER^{TM} 1.1 from Bruker Daltonik, Bremen, Germany. The involved peak picking generates a peak list vector for each spectrum. All peak list vectors are aggregated such that finally a data matrix with 937 intensity components and 109 rows is achieved as data base. Thereby, a peak shift tolerance of 300ppm was used. A classification tree obtained by hierarchical clustering using the BIOTYPER^{TM} 1.1 software yields a class separation tree as depicted in Fig.1.

This tree can serve for comparison for FLSOM class similarity detection.

3.1.2 The breast cancer tissue data set

The breast cancer tissues are collected at the Institute of Pathology in Neuherberg, GSF-National Research Center for Environment an Health, Germany. The generic measurement procedure can be summarized as follows. The frozen tissue is cut using a cryomicrotome

Figure 1: Separation tree of different listeria types obtained by BIOTYPER^{TM} 1.1 software based on hierarchical clustering.

in sections of 12μm and transferred to a conductive slide, washed in ethanol and coated by matrix. Reference sections are used for histological staining (Hematoxylin&Eosin, immunohistochemical staining for HER2) and histomorphological classification. The slices are subsequently measured in a *Ultraflex II* MALDI-TOF and subsequently visualized using the FlexImaging^{TM} tool provided by Bruker Daltonik, Bremen, Germany [12].

The breast cancer tissue slices are manually labeled by a clinical expert (pathologist). In this exemplary study the slice of one patient is used. Four different spatial regions of the slice are marked according to histomorphologically classfied cell types: connective tissue, inflammation, and two morphologically distinct tumor cell populations (tumor-type-1, tumor-type-2). From the whole slice 687 spectral record are generated, 438 of them are labeled with at least 51 records per region.

These profiles were preprocessed (baseline correction, alignment, peak picking and, peak feature extraction by means of maximum intensities) according to the CLINPROTOOLS^{TM} 2.1 from Bruker Daltonik, Bremen, Germany, see [11]. Finally, each preprocessed data record is a 70-dimensional data vector.

3.2 Application of the methods and interpretation of the results

Both data sets are analyzed by SNG and FLSOM using the scaled quadratic Euclidean metric as data similarity measure. For comparison we also applied SVM with different kernels and LDA based on linear PCA. Additionally, for the bacteria data set a *class dependence tree* provided as standard solution of the BIOTYPER^{TM} 1.1 tool based on hierarchical clustering is also available for comparison [20].

In case of FLSOM application we further investigate the detected class similarities and provide visualization results. The optimum FLSOM lattice size was estimated by a GSOM using standard Euclidean metric for data.

3.2.1 Results for the bacteria data set

First, we applied SNG with 3 prototypes per class and the neighborhood range σ for regularization is slowly decreased to zero during the adaptation process. For FLSOM a two-dimensional 12×4 lattice structure is suggested by the GSOM. Due to the large number of prototypes for FLSOM in comparison to SNG and paying attention to the sparse data the final regularization parameter for FLSOM is set to $\sigma = 0.4$, which yields non-vanishing regularization. The balancing parameter was set to $\beta = 0.05$ in the beginning and increasung up to the final value of $\beta = 0.85$. The topology preservation of the FLSOM is preserved, as the topographic product value $TP = -0.0066$ indicates. The 5-fold cross-validated classification

LDA	SVM_1	SVM_2	SVM_3	SNG	FLSOM
61.5%	34.9%	$< 10\%$	96.3%	97.8%	73.4%

Table 1: Classification accuracies for the different classifiers for the listeria data set. SVM_1 is a linear SVM, SVM_2 uses a radial basis function kernel and SVM_3 uses a Tanimoto-distance-kernel.LDA is based on linear PCA suggesting 5 principal components to be sufficient. For FLSOM majority vote is taken to obtain the crisp classification.

accuracy results are collected in Tab. 1. Both, SNG and FLSOM show very good performance in comparison to the other algorithms. In particular, we remark that SVM heavily depends on the used kernel, as it is also known from other applications [25]. The slightly decreased accuracy of the FLSOM is the consequence of the balancing parameter $\beta < 1.0$, which is necessary for stability reasons of the algorithm as described before. However, this disadvantage is compensated by the class similarity detection feature provided by FLSOM [36]. These results are now under deeper consideration. The fuzzy class label vectors $\mathbf{y_r}$ of the prototypes are depicted in Fig. 2a) according to their distribution with respect to the FLSOM lattice.

This distribution of the prototype labels suggests the following interpretation of FLSOM-detected class similarities, which should also be compared to the above given separation tree obtained by the BIOTYPER^{TM} 1.1 software depicted in Fig 1: The listeria of grayi types (class 1 & 2) should not be distinguished according to their proteom finger print. The class 8 (listeria welshimeri) is clearly isolated from each other. Classes 4, 5 and 7 (listeria ivanovii ssp ivanovii, listeria seeligeri and listeria ivanovii ssp londoniensis) are very similar. Further there is a class similarity between the classes 3 and 4 (listeria innocua and listeria ivanovii ssp ivanovii). Although this similarity in the tree classification can not be ruled out, the tree visualization suggests a stronger separation. This 'separation' would be disappear, if the respective branch would be rotated. Thus, the FLSOM label distribution is more adequate. Further, FLSOM detected a similarity between the classes 6 and 4 (listeria monocytogenes and listeria ivanovii ssp ivanovii), which is also not easily detectable in the tree visualization. The class 3 (listeria innocua) shows multiple similarities to several other species based on the proteom finger print. This sharing property can not be reflected adequately in the tree classification. However, an expert biologist independently suggested a similarity between both types[2].

Thus summerizing, the FLSOM provides detailed class similarity description together with comparable classification accuracy, whereas SNG achieves best accuracy.

3.2.2 Results for the breast cancer tissue data set

Again, we applied SNG with 3 prototypes per class and the neighborhood range σ for regularization also slowly decreasing to zero during the adaptation process. For FLSOM a two-dimensional 15×5 lattice structure is suggested here by the GSOM. As for the listeria data set the final regularization parameter for FLSOM is set to $\sigma = 0.4$, which yields non-vanishing regularization. The balancing parameter setting was the same as for the bacteria data set. The topographic product value $TP = 0.0001$ ensures the topographic mapping of the FLSOM. The 5-fold cross-validated classification accuracy results are collected in Tab. 2.

[2]Personal communication with Dr. Thomas Maier, BRUKER Daltonik Leipzig, Germany.

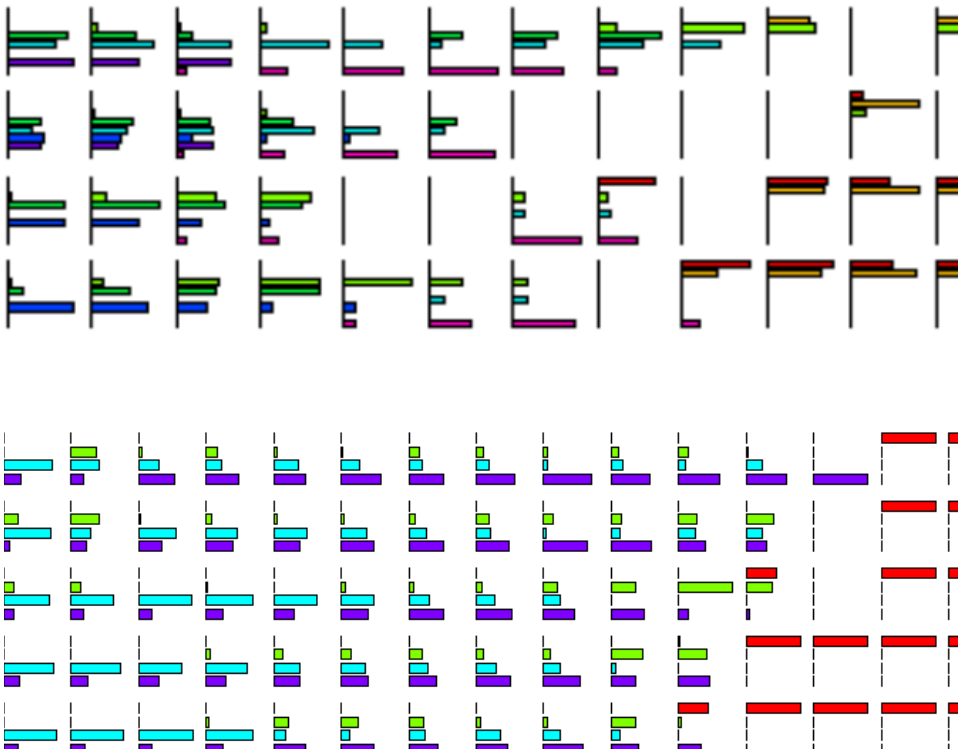


Figure 2: Distribution of the class responsibilities within the FLSOM-lattices for a) the listeria classification problem and b) the breast cancer problem. The label vectors $\mathbf{y_r}$ are depicted as barplots arranged according to the FLSOM-grid structure. Each barplot refers to a label vector, whereby the height of the bars within is according to the probability that the prototype is responsible for the respective class (left class 1 – right class 8, 4 - respectively). The coloring of the bars is only for better visualization and does not contain any information. Flat lines show 'dead' prototypes, i.e. which did not won the competing process for the available data.

LDA	SVM_1	SVM_2	SVM_3	SNG	FLSOM
59.8%	62.8%	42.7%	84.2%	80.4%	72.4%

Table 2: Classification accuracies for the different classifiers for the breast cancer data set. SVM_1 is a linear SVM, SVM_2 uses a radial basis function kernel and SVM_3 uses a Tanimoto-distance-kernel.LDA is based on linear PCA suggesting 8 principal components to be sufficient. For FLSOM majority vote is taken to obtain the crisp classification.

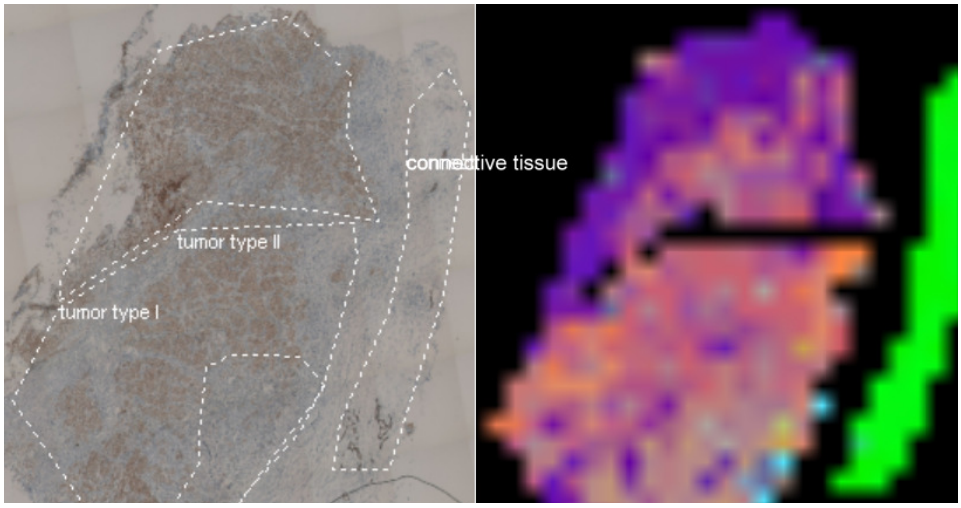


Figure 3: Breast cancer tissue section with manually labeled areas used for classification training is depicted left. Right hand, the classification obtained by the FLSOM classifier is plotted using an MDS RGB-color embedding of the FLSOM-label vectors $\mathbf{y_r}$. Thereby, similar colors represent similar class properties as detected by FLSOM (black - not used for classification). One clearly see the fine agreement with the manual labelling.

The obtained accuracies for SNG and FLSOM show high levels with a slightly decreased value for FLSOM, whereas SNG achieves the overall best result.

For class similarity investigation we consider the distribution of the label vectors within the FLSOM-grid, which is depicted in Fig. 2. The connective-tissue class is well separated. Further, we have in the distribution plane an overlapping region between the both tumor classes, which indicates a similarity between them. The FLSOM detects a clear distinction between tumor-1-class and the connective tissue class according to the spatial distribution of the respective labels in the FLSOM grid, whereas small overlapping between tumor-2-class and the connective-tissue class occurs. The inflammation class shows similarity to type-2-tumor. Using an MDS color embedding of all label vectors $\mathbf{y_r}$ of the FLSOM into the RGB-color space, the FLSOM classification of the tissue can be easily visualized, see Fig 3.

A high agreement between original manually labeled tissue and obtained coloring based on the classification and class similarity detection generated by the FLSOM can be observed.

Further information can be obtained by consideration of the learned relevance profile of the problem specific scaled Euclidian metric. It is depicted in Fig.4.

The highest relevance for class separation can be assigned to the 4971Da-peak in the original proteomic spectra. Recoloring of the original tissue according to the 4971Da-intensities shows that this peak mainly separates the connective tissue class from the other classes, see

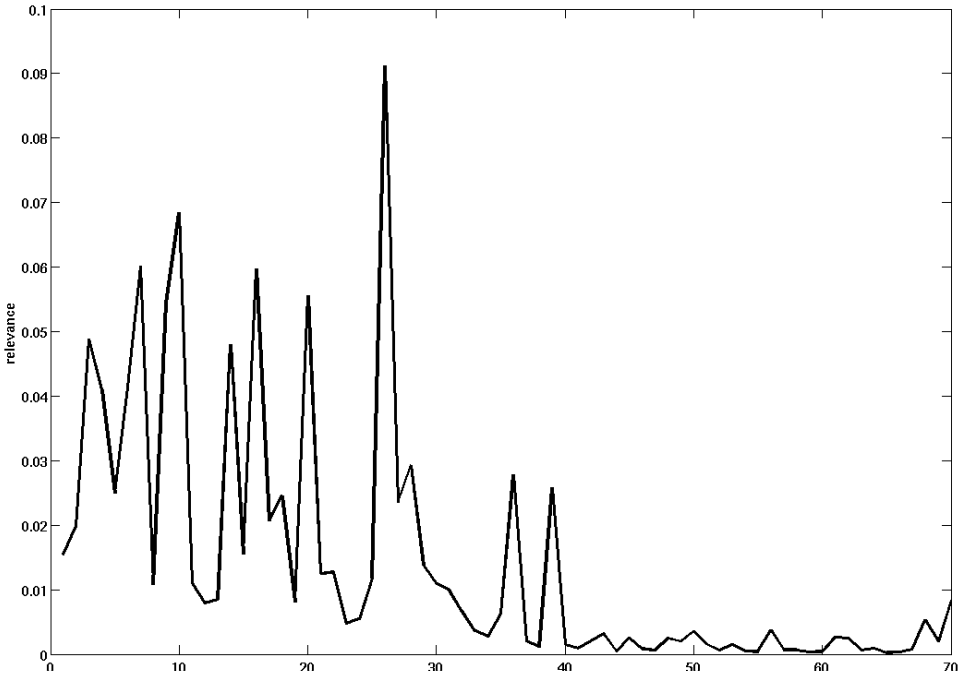


Figure 4: Relevance profile of the scaled Euclidian metric for the breast cancer problem. The highest relevance peak can be assigned to the data dimension 26 which is assigned to the 4971Da-peak in the original proteomic spectrum.

Fig 5.

Analogously, the other relevance peaks could be evaluated. In this way, a detailed analysis of the information contained in the FLSOM model can be obtained.

4 Concluding remarks

In this article two recently developed prototype based methods for classification, SNG and FLSOM, are reviewed in the light of the analysis of mass spectrometric data in bioninformatics. Both approaches are adaptive machine learning approaches and allow easy retraining, if new data become available. They are both inherently regularizing, such that they are able to handle sparse, high-dimensional and noisy data. As demonstrated for two exemplary problems in classification of proteomic spectra (bacteria and breast cancer tissue), the generated classification models show good performance compared to other machine learning and statistical methods.

Additionally, FLSOM provides the possibility of processing uncertain class information for training data (fuzzy) and returns a fuzzy classification scheme. Moreover, FLSOM provides a class similarity detection based on the fuzzy labels, which give the possibility of deeper class analysis offering more information than simple classification trees. The fuzzy classification can further be used for class dependent data visualization whereby similar class information is encoded by similar colors such that an easy interpretation can be made.

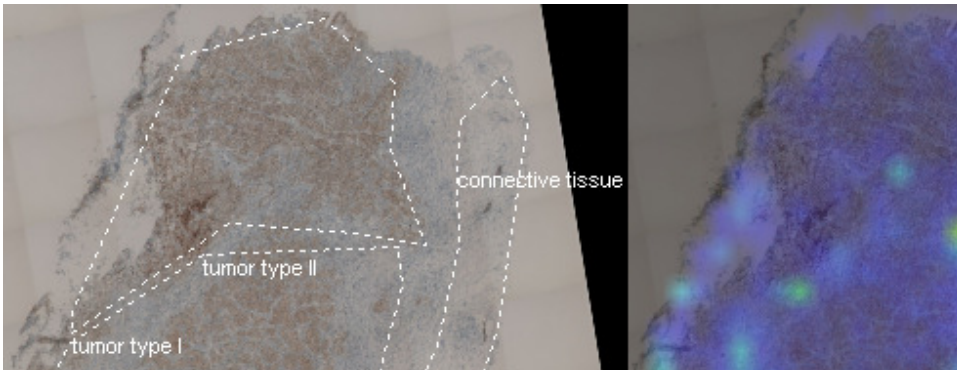


Figure 5: Recoloring of the original tissue according to the 4971Da-intensities. High intensities are colored red, low values are coded by blue colors.

References

[1] P. Baldi and S. Brunak. *Bioinformatics – The Machine Learning Approach.* The MIT Press, 1998.

[2] H.-U. Bauer and K. R. Pawelzik. Quantifying the neighborhood preservation of Self-Organizing Feature Maps. *IEEE Trans. on Neural Networks*, 3(4):570–579, 1992.

[3] H.-U. Bauer and T. Villmann. Growing a Hypercubical Output Space in a Self–Organizing Feature Map. *IEEE Transactions on Neural Networks*, 8(2):218–226, 1997.

[4] C. Bishop. *Pattern Recognition and Machine Learning.* Springer Science+Business Media, LLC, New York, NY, 2006.

[5] C. M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, 1995.

[6] K. Crammer, R. Gilad-Bachrach, A.Navot, and A.Tishby. Margin analysis of the LVQ algorithm. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing (Proc. NIPS 2002)*, volume 15, pages 462–469, Cambridge, MA, 2003. MIT Press.

[7] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University Press, 2000.

[8] M. de Noo, A. Deelder, M. van der Werff, A. zalp, and B. Martens. MALDI-TOF serum protein profiling for detection of breast cancer. *Onkologie*, 29:501–506, 2006.

[9] M. de Noo, B. Martens, A. zalp, M. Bladergroen, M. van der Werff, C. van de Velde A. Deelder, and R. Tollenaar. Detecting of colorectal cancer using MALDI-TOF serum protein profiling. *European Journal of Cancer*, 42:1068–1076, 2006.

[10] R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, 1973.

[11] M. Gerhard, S.-O. Deininger, and F.-M. Schleif. Statistical classification and visualization of MALDI-imaging data. In P. Kokol, M. Zorman, V. Podgerelec, M. Verlic, and

D. Micetic-Turk, editors, *Proceedings of the 20th IEEE Symposium on Computer-based Medical Systems*, pages 403–405. IEEE Press, 2007.

[12] M. Groseclose, M. Andersson, W. Hardesty, and R. Caprioli. Identifications of proteins directly from tissue: in situ tryptic digestions coupled with imaging mass spectrometry. *Journal of Mass Spectrometry*, 42:254–262, 2007.

[13] B. Hammer, M. Strickert, and T. Villmann. Supervised neural gas with general similarity measure. *Neural Processing Letters*, 21(1):21–44, 2005.

[14] B. Hammer and T. Villmann. Generalized relevance learning vector quantization. *Neural Networks*, 15(8-9):1059–1068, 2002.

[15] B. Hammer and T. Villmann. Classification using non-standard metrics. In M. Verleysen, editor, *Proc. Of European Symposium on Artificial Neural Networks (ESANN'2005)*, pages 303–316, Brussels, Belgium, 2005. d-side publications.

[16] T. Heskes. Energy functions for self-organizing maps. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 303–316. Elsevier, Amsterdam, 1999.

[17] R. Ketterlinus, S.-Y. Hsieh, S.-H. Teng, H. Lee, and W. Pusch. Fishing for biomarkers: analyzing mass spectrometry data with the new clinprotools software. *Bio techniques*, 38(6):37–40, 2005.

[18] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 1995. (Second Extended Edition 1997).

[19] J. Lee and M. Verleysen. Generalization of the l_p norm for time series and its application to self-organizing maps. In M. Cottrell, editor, *Proc. of Workshop on Self-Organizing Maps (WSOM) 2005*, pages 733–740, Paris, Sorbonne, 2005.

[20] T. Maier and M. Kostrzewa. Fast and reliable MALDI-TOF MS-based microorganism identification. *Chemistry Today*, 25(2):68–71, 2007.

[21] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten. 'Neural-gas' network for vector quantization and its application to time-series prediction. *IEEE Trans. on Neural Networks*, 4(4):558–569, 1993.

[22] A. Sato and K. Yamada. Generalized learning vector quantization. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8. Proceedings of the 1995 Conference*, pages 423–9. MIT Press, Cambridge, MA, USA, 1996.

[23] F.-M. Schleif, T. Elssner, M. Kostrzewa, T. Villmann, and B. Hammer. Analysis and visualization of proteomic data by fuzzy labeled self-organizing maps. In D. Lee, B. Nutter, S. Antani, S. Mitra, and J. Archibald, editors, *19th IEEE International Symposium on Computer- based Medical Systems Salt Lake City (CBMS)*, pages 919–924. IEEE Computer Society Press, Los Alamitos, 2006. 0769525171.

[24] J. Schürmann. *Pattern Classification*. J. Wiley and Sons Inc., New York, 1996.

[25] B. Schlkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

[26] U. Seiffert, B. Hammer, S. Kaski, and T. Villmann. Neural networks and machine learning in bioinformatics - theory and applications. In M. Verleysen, editor, *Proc. Of European Symposium on Artificial Neural Networks (ESANN'2006)*, pages 521–532, Brussels, Belgium, 2006. d-side publications.

[27] U. Seiffert, L. C. Jain, and P. Schweizer. *Bioinformatics using Computational Intelligence Paradigms*. Springer-Verlag, 2004.

[28] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis and Discovery*. Cambridge University Press, 2004.

[29] M. Strickert, U. Seiffert, N. Sreenivasulu, W. Weschke, T. Villmann, and B. Hammer. Generalized relevance LVQ (GRLVQ) with correlation measures for gene expression analysis. *Neurocomputing*, 69(6–7):651–659, March 2006. ISSN: 0925-2312.

[30] S. Swamidass and P. Baldi. Bounds and algorithms for fast exact searches of chemical fingerprints in linear and sublinear time. *Journal of Chemical Information and Modeling*, 47(2):302–317, 2007.

[31] K. Todar. Todar's online textbook of bacteriology – listeria monocytogenes and listerisis. Univsersity of Wisconsin-Madion Department of Biology, http://textbookofbacteriology.net/Listeria.html, 2003.

[32] M. Verleysen and D. François. The curse of dimensionality in data mining and time series prediction. In J. Cabestany, A. Prieto, and F. S. Hernández, editors, *Computational Intelligence and Bioinspired Systems, Proceedings of the 8th International Work-Conference on Artificial Neural Networks 2005 (IWANN), Barcelona*, pages 758–770, Berlin, 2005. Springer.

[33] T. Villmann and H.-U. Bauer. Applications of the growing self-organizing map. *Neurocomputing*, 21(1-3):91–100, 1998.

[34] T. Villmann, R. Der, M. Herrmann, and T. Martinetz. Topology Preservation in Self–Organizing Feature Maps: Exact Definition and Measurement. *IEEE Transactions on Neural Networks*, 8(2):256–266, 1997.

[35] T. Villmann, F.-M. Schleif, and B. Hammer. Comparison of relevance learning vector quantization with other metric adaptive classification methods. *Neural Networks*, 19:610–622, 2006.

[36] T. Villmann, F.-M. Schleif, E. Merényi, M. Strickert, and B. Hammer. Class imaging of hyperspectral satellite remote sensing data using flsom. In H. Ritter, editor, *Proc. Workshop on Self-Organizing Maps WSOM*, page in press. Bielefeld, Germany, 2007.

[37] T. Villmann, U. Seiffert, F.-M. Schleif, C. Brüß, T.Geweniger, and B. Hammer. Fuzzy labeled self-organizing map with label-adjusted prototypes. In F. Schwenker and S. Marinai, editors, *Proceedings of Conference Artificial Neural Networks in Pattern Recognition (ANNPR) 2006, Ulm, Germany*, LNAI 4087, pages 46–56. Springer Verlag, 2006.

[38] T. Villmann, M. Strickert, C. Brüß, F.-M. Schleif, and U. Seiffert. Visualization of fuzzy information in fuzzy-classification for image segmentation using MDS. In M. Verleysen, editor, *Proc. Of European Symposium on Artificial Neural Networks (ESANN'2007)*, pages 103–108, Brussels, Belgium, 2007. d-side publications.

2.3 Prototype based Fuzzy Classification in Clinical Proteomics

In the article *Prototype based Fuzzy Classification in Clinical Proteomics* the fuzzy concept for prototype based learning was employed for Soft-Nearest Prototype Classification and compared to Fuzzy Labeled Neural Gas. The approaches where analyzed as *local* models, using *relevance learning* in the context of clinical proteomics. The article was written by F.-M.**Schleif**, T. Villmann, and B. Hammer. It appeared in 2008 in the International Journal of Approximate Reasoning, 47(1), p. 4-16. I wrote the main parts of the manuscript and implemented the methods, by combining multiple concepts published by the authors priorly. I also run the experiments and prepared the data sets. Thomas Villmann and Barbara Hammer supervised the project. All authors discussed the paper.

Prototype based Fuzzy Classification in Clinical Proteomics

F.-M. Schleif[1]*, T. Villmann[2], B. Hammer[3]
[1]Bruker Daltonik GmbH Leipzig and Univ. Leipzig, Dept. of CS
[2]University Leipzig, Clinic for Psychotherapy
[3]Clausthal Univ. of Tech., Dept. of Math. and CS

Abstract

Proteomic profiling based on mass spectrometry is an important tool for studies at the protein and peptide level in medicine and health care. Thereby, the identification of relevant masses, which are characteristic for specific sample states e.g. a disease state is complicated. Further, the classification accuracy and safety is especially important in medicine. The determination of classification models for such high dimensional clinical data is a complex task. Specific methods, which are robust with respect to the large number of dimensions and fit to clinical needs, are required. In this contribution two such methods for the construction of nearest prototype classifiers are compared in the context of clinical proteomic studies, which are specifically suited to deal with such high-dimensional functional data. Both methods are suitable to the adaptation of the underling metric, which is useful in proteomic research to get a problem adequate representation of the clinical data. In addition they allow fuzzy classification and for one of them allows fuzzy classified training data. Both algorithms are investigated in detail with respect to their specific properties. A performance analyzes is taken on real clinical proteomic cancer data in a comparative manner.

Keyword: fuzzy classification, learning vector quantization, metric adaptation, mass spectrometry, proteomic profiling

1 Introduction

During last years proteomic[1] profiling based on mass spectrometry (MS) became an important tool for studying cancer at the protein and peptide level in a high throughput manner. MS based serum profiling is under development as a potential diagnostic tool to distinguish between patients suffering from cancer and healthy subjects. Reliable classification methods, which can cope with typically high-dimensional characteristic profiles, constitute a crucial part of the system. Thereby, a good generalization ability and interpretability of the results are highly desirable. Prototype based classification is intuitive approach based on representatives (prototypes) for the respective classes.

Kohonen's Learning Vector Quantization (LVQ) belongs to the class of supervised learning algorithms for nearest prototype classification (NPC) [2]. It relies on a set of prototype

* *corresponding author,Bruker Daltonik GmbH, Permoserstrasse 15, D-04318 Leipzig, Germany,Tel: +49 341 24 31-408, Fax: +49 341 24 31-404,email: fms@bdal.de*

[1]Proteome - is an ensemble of protein forms expressed in a biological sample at a given point in time [1].

vectors (also called codebook vectors), which are adapted by the algorithm according to their respective classes. Thus, it forms a very intuitive local classification method with very good generalization ability also for high-dimensional data [3], which constitutes an ideal candidate for an automatic and robust classification tool for high throughput proteomic patterns.

However, original LVQ is only heuristically motivated and shows instable behavior for overlapping classes. Recently a new method, Soft Nearest Prototype Classification (SNPC), has been proposed by Seo et al. [4] based on the formulation as a Gaussian mixture approach, which yields soft assignments of data. This algorithm can be extended by local and global metric adaptation (called relevance learning) to (L)SNPC-R [5] and applied in profiling of mass spectrometric data in cancer research. In addition, the learning of the prototype labels has been changed to support fuzzy values, which finally allows fuzzy prototype labels yielding fuzzy SNPC (FSNPC) [6]. The approach is well suited to deal with high-dimensional data focusing on optimal class separability. Further, it is capable to determine relevance profiles of the input, which can be used for identification of relevant data dimensions. In addition, the metric adaptation parameters may be further analyzed with respect to clinical knowledge extraction.

The second algorithm also refers to the class of LVQ networks but was originally motivated as an unsupervised clustering approach, named Neural GAS introduced in [7]. This algorithm distributes the prototypes such that the data density is estimated by minimizing some description error aiming at unsupervised data clustering. Prototype based classification as a supervised vector quantization scheme is dedicated to distribute prototypes in such a manner that data classes can be detected, which naturally is influenced by the data density, too. Taking this into account the Fuzzy Labeled Neural GAS algorithm (FLNG) has been introduced in [8, 9]. This algorithm will be used as a second prototype based classification approach in this contribution. The capabilities of different variants of FSNPC and FLNG are demonstrated for different cancer data sets: the Wisconsin Breast Cancer (WBC)[10], the leukemia data set (LEUK) provided by [11] and two other non-public proteomic data obtained from [12].

The paper is organized as follows: the crisp SNPC is reviewed in section 2 followed by the extension of metric adaptation (relevance learning (SNPC-R)). Thereafter the concept of fuzzy classification is derived for the SNPC algorithm and also combined with the relevance concept. In section 3 the FLNG algorithm will be presented. Subsequently, application results of the algorithms are reported in a comparative manner. The article concludes by a short discussion of the methods and shows the benefits of the metric adaptation as well as of fuzzy classification for clinical data.

2 Soft nearest prototype classification

Usual learning vector quantization is a prototype based classification methodology, mainly influenced by the standard algorithms LVQ1...LVQ3 introduced by Kohonen [2]. Several derivatives have been developed to ensure faster convergence, a better adaptation of the receptive fields to optimum Bayesian decision, or an adaptation for complex data structures [13, 14, 4]. Any of the above algorithms LVQ1...LVQ3, does not possess a cost function in the continuous case; it is based on the heuristic to minimize misclassifications using Hebbian learning. The first version of learning vector quantization based on a cost function, which formally assesses the misclassifications, is the Generalized LVQ (GLVQ) [15]. GLVQ resp. its extensions Supervised Neural GAS (SNG) and Supervised Relevance Neural GAS (SRNG) as introduced in [16] will be used for comparison in this article.

First, basic notations for LVQ schemes are introduced. Inputs are denoted by $\mathbf{v}$ with

label $c_{\mathbf{v}} \in \mathcal{L}$. Assume $\mathcal{L}$ is the set of labels (classes) with $\#\mathcal{L} = N_{\mathcal{L}}$ and $V \subseteq \mathbb{R}^{D_V}$ a finite set of inputs $\mathbf{v}$. LVQ uses a fixed number of prototypes (weight vectors, codebook vectors) for each class. Let $\mathbf{W} = \{\mathbf{w}_{\mathbf{r}}\}$ be the set of all codebook vectors and $c_{\mathbf{r}}$ be the class label of $\mathbf{w}_{\mathbf{r}}$. Furthermore, let $\mathbf{W}_c = \{\mathbf{w}_{\mathbf{r}} | c_{\mathbf{r}} = c\}$ be the subset of prototypes assigned to class $c \in \mathcal{L}$. The classification of vector quantization is implemented by the map Ψ as a winner-take-all rule, i.e. a stimulus vector $\mathbf{v} \in V$ is mapped onto that neuron $\mathbf{s} \in A$ the pointer $\mathbf{w}_s$ of which is closest to the presented vector $\mathbf{v}$,

$$\Psi_{\mathcal{V}\to\mathcal{A}} : \mathbf{v} \mapsto \mathbf{s}(\mathbf{v}) = \underset{\mathbf{r}\in A}{\operatorname{argmin}}\, d(\mathbf{v}, \mathbf{w}_{\mathbf{r}}) \tag{2.1}$$

with $d(\mathbf{v}, \mathbf{w})$ being an arbitrary distance measure, usually the squared euclidean metric. The neuron $\mathbf{s}$ is called winner or best matching unit. The subset of the input space $\Omega_{\mathbf{r}} = \{\mathbf{v} \in V : \mathbf{r} = \Psi_{V\to A}(\mathbf{v})\}$, which is mapped to a particular neuron $\mathbf{r}$ according to (2.1), forms the (masked) receptive field of that neuron. Standard LVQ training adapts the prototypes such that for each class $c \in \mathcal{L}$, the corresponding codebook vectors $\mathbf{W}_c$ represent the class as accurately as possible, i.e. the set of points in any given class $V_c = \{\mathbf{v} \in V | c_{\mathbf{v}} = c\}$, and the union $\mathcal{U}_c = \bigcup_{\mathbf{r}|_{\mathbf{w}_{\mathbf{r}}\in\mathbf{W}_c}} \Omega_{\mathbf{r}}$ of receptive fields of the corresponding prototypes should differ as little as possible. This is either achieved by heuristics as for LVQ1...LVQ3 [2], or by the optimization of a cost function related to the mismatches as for GLVQ [15] and SRNG as introduced in [16].

Soft Nearest Prototype Classification (SNPC) has been proposed as alternative stable NPC learning scheme. It introduces soft assignments for data vectors to the prototypes, which have a statistical interpretation as normalized Gaussians. In the original SNPC as provided in [4] one considers

$$E(\mathcal{S}) = \frac{1}{N_{\mathcal{S}}} \sum_{k=1}^{N_{\mathcal{S}}} \sum_{\mathbf{r}} u_\tau(\mathbf{r}|\mathbf{v}_k) \left(1 - \alpha_{\mathbf{r},c_{\mathbf{v}_k}}\right) \tag{2.2}$$

as the cost function with $\mathcal{S} = \{(\mathbf{v}, c_{\mathbf{v}})\}$ the set of all input pairs, $N_{\mathcal{S}} = \#\mathcal{S}$. The class assignment variables $\alpha_{\mathbf{r},c_{\mathbf{v}_k}}$ equals one if $c_{\mathbf{v}_k} = c_{\mathbf{r}}$ and 0 otherwise, i.e. the assignments are crisp. $u_\tau(\mathbf{r}|\mathbf{v}_k)$ is the probability that the input vector $\mathbf{v}_k$ is assigned to the prototype $\mathbf{r}$. A crisp *winner-takes-all* mapping (2.1) would yield $u_\tau(\mathbf{r}|\mathbf{v}_k) = \delta(\mathbf{r} = \mathbf{s}(\mathbf{v}_k))$.

In order to minimize (2.2), in [4] the variables $u_\tau(\mathbf{r}|\mathbf{v}_k)$ are taken as soft assignment probabilities. This allows a gradient descent on the cost function (2.2). As proposed in [4], the probabilities (soft assignments) are chosen as normalized Gaussians

$$u_\tau(\mathbf{r}|\mathbf{v}_k) = \frac{\exp\left(-\frac{d(\mathbf{v}_k,\mathbf{w}_{\mathbf{r}})}{2\tau^2}\right)}{\sum_{\mathbf{r}'} \exp\left(-\frac{d(\mathbf{v}_k,\mathbf{w}_{\mathbf{r}'})}{2\tau^2}\right)} \tag{2.3}$$

whereby d is the distance measure used in (2.1) and τ is the bandwidth which has to be chosen adequately. Then the cost function (2.2) can be rewritten as

$$E(\mathcal{S}) = \frac{1}{N_{\mathcal{S}}} \sum_{k=1}^{N_{\mathcal{S}}} lc((\mathbf{v}_k, c_{\mathbf{v}_k})) \tag{2.4}$$

with local costs

$$lc((\mathbf{v}_k, c_{\mathbf{v}_k})) = \sum_{\mathbf{r}} u_\tau(\mathbf{r}|\mathbf{v}_k) \left(1 - \alpha_{\mathbf{r},c_{\mathbf{v}_k}}\right) \tag{2.5}$$

i.e., the local error is the sum of the class assignment probabilities $\alpha_{\mathbf{r},c_{\mathbf{v}_k}}$ to all prototypes of an incorrect class, and, hence

$$lc\left((\mathbf{v}_k, c_{\mathbf{v}_k})\right) \leq 1 \tag{2.6}$$

with local costs depending on the whole set $\mathbf{W}$. Because the local costs $lc\left((\mathbf{v}_k, c_{\mathbf{v}_k})\right)$ are continuous and bounded, the cost function (2.4) can be minimized by stochastic gradient descent using the derivative of the local costs:

$$\triangle \mathbf{w}_\mathbf{r} = \begin{cases} \frac{1}{2\tau^2} u_\tau\left(\mathbf{r}|\mathbf{v}_k\right) \cdot lc\left((\mathbf{v}_k, c_{\mathbf{v}_k})\right) \cdot \frac{\partial d_\mathbf{r}}{\partial \mathbf{w}_\mathbf{r}} & \text{if} \quad c_{\mathbf{v}_k} = c_\mathbf{r} \\ -\frac{1}{2\tau^2} u_\tau\left(\mathbf{r}|\mathbf{v}_k\right) \cdot \left(1 - lc\left((\mathbf{v}_k, c_{\mathbf{v}_k})\right)\right) \cdot \frac{\partial d_\mathbf{r}}{\partial \mathbf{w}_\mathbf{r}} & \text{if} \quad c_{\mathbf{v}_k} \neq c_\mathbf{r} \end{cases} \tag{2.7}$$

where

$$\frac{\partial lc}{\partial \mathbf{w}_\mathbf{r}} = -u_\tau\left(\mathbf{r}|\mathbf{v}_k\right) \left(\left(1 - \alpha_{\mathbf{r},c_{\mathbf{v}_k}}\right) - lc\left((\mathbf{v}_k, c_{\mathbf{v}_k})\right)\right) \cdot \frac{\partial d_\mathbf{r}}{\partial \mathbf{w}_\mathbf{r}} \tag{2.8}$$

This leads to the learning rule

$$\mathbf{w}_\mathbf{r} = \mathbf{w}_\mathbf{r} - \epsilon\left(t\right) \cdot \triangle \mathbf{w}_\mathbf{r} \tag{2.9}$$

with learning rate $\epsilon\left(t\right)$ fulfilling $\sum_{t=0}^{\infty} \epsilon\left(t\right) = \infty$ and $\sum_{t=0}^{\infty} \left(\epsilon\left(t\right)\right)^2 < \infty$ as usual. All prototypes are adapted in this scheme according to the soft assignments. Note that for small bandwidth τ, the learning rule is similar to LVQ2.1.

A window rule like for standard LVQ2.1 can be derived for SNPC, too, which is necessary for numerical stabilization [2],[4]. The update is restricted to all weights for which the local value

$$\eta = lc\left((\mathbf{v}_k, c_{\mathbf{v}_k})\right) \cdot \left(1 - lc\left((\mathbf{v}_k, c_{\mathbf{v}_k})\right)\right) \tag{2.10}$$

is less than a threshold value η with $0 \ll \eta < 0.25$ [4]. The justification for this fact is given in [4] (page 4).

2.1 Relevance learning for SNPC

Like all NPC algorithms, SNPC heavily relies on the metric d, usually the standard euclidean metric. For high-dimensional data as occur in proteomic patterns, this choice is not adequate since noise present in the data set accumulates and likely disrupts the classification. Thus, a focus on the (priory not known) relevant parts of the inputs, would be much more suited. Relevance learning as introduced in [17] offers the opportunity to learn metric parameters, which is called relevance learning. This concept now is included into the above SNPC and well be referred as SNPC-R: A parameter vector $\lambda = (\lambda_1, \ldots, \lambda_m)$ is assigned to the metric $d\left(\mathbf{v}_k, \mathbf{w}_\mathbf{r}\right)$ denoted as $d^\lambda\left(\mathbf{v}_k, \mathbf{w}_\mathbf{r}\right)$, which now is used in the soft assignments (2.3). One popular example is the scaled Euclidean metric

$$d^\lambda\left(\mathbf{v}_k, \mathbf{w}_\mathbf{r}\right) = \sum_{i=1}^{D_V} \lambda_i (\mathbf{v}_k^i - \mathbf{w}_\mathbf{r}^i)^2. \tag{2.11}$$

Parallelly to the usual prototype adaptation the relevance parameters λ_j can be adjusted according to the given classification problem, taking the respective derivative of the cost function. Doing so the derivative of the local costs (2.5) becomes

$$\frac{\partial lc\left((\mathbf{v}_k, c_{\mathbf{v}_k})\right)}{\partial \lambda_j} = \frac{1}{2\tau^2} \sum_\mathbf{r} u_\tau\left(\mathbf{r}|\mathbf{v}_k\right) \cdot \frac{\partial d_\mathbf{r}^\lambda}{\partial \lambda_j} \cdot \left(\alpha_{\mathbf{r},c_{\mathbf{v}_k}} + lc\left((\mathbf{v}_k, c_{\mathbf{v}_k})\right) - 1\right) \tag{2.12}$$

followed by a subsequent normalization of the λ_j.

It is worth to emphasize that SNPC-R can also be used with *individual* metric parameters $\lambda^{\mathbf{r}}$ for each prototype $\mathbf{w_r}$ or with a classwise metric shared within prototypes with the same class label c_r as it is done here, referred as localized SNPC-R (LSNPC-R). If the metric is shared by all prototypes, LSNPC-R is reduced to SNPC-R. The respective adjusting of the relevance parameters λ can easily be determined in complete analogy to (2.12).

It has been pointed out in [3] that NPC classification schemes, which are based on the euclidean metric, can be interpreted as large margin algorithms for which dimensionality independent generalization bounds can be derived. Instead of the dimensionality of data, the so-called hypothesis margin, i.e. the distance, the hypothesis can be altered without changing the classification on the training set, serves as a parameter of the generalization bound. This result has been extended to NPC schemes with *adaptive* diagonal metric in [16]. This fact is quite remarkable, since D_V new parameters, D_V being the input dimension, are added this way, still, the bound is independent of D_V. This result can even be transferred to the setting of *individual* metric parameters $\lambda^{\mathbf{r}}$ for each prototype or class such that a generally good generalization ability of this method can be expected [18]. Despite from the fact that (possibly local) relevance factors allow a larger flexibility of the approach without decreasing the generalization ability,they are of particular interest for proteomic pattern analysis because they indicate potentially semantically meaningful positions.

2.2 Fuzzy classification for SNPC-R

In *Fuzzy Labeled* SNPC (FSNPC) one now allows fuzzy values for $\alpha_{\mathbf{r},c}$ to indicate the responsibility of weight vector $\mathbf{w_r}$ to class c such that now

$$0 \leq \alpha_{\mathbf{r},c} \leq 1$$

in contradiction to the crisp case and under the normalization condition $\sum_{c=1}^{N_{\mathcal{L}}} \alpha_{\mathbf{r},c} = 1$. These labels should be adjusted *automatically* during training. However, doing so, the crisp class information for prototypes, assumed in the learning dynamic of SNPC (2.7) (or generally required in LVQ) [4], is no longer available. However, a corresponding learning dynamic can be derived: In complete analogy to the original SNPC with the same cost function (2.4) one gets

$$\triangle \mathbf{w_r} = -\frac{T}{2\tau^2} \cdot \frac{\partial d_{\mathbf{r}}}{\partial \mathbf{w_r}} \tag{2.13}$$

with

$$T = u_\tau\left(\mathbf{r}|\mathbf{v}_k\right) \cdot \left(1 - \alpha_{\mathbf{r},c_{\mathbf{v}_k}} - lc\left(\mathbf{v}_k, c_{\mathbf{v}_k}\right)\right).$$

Thereby, the loss boundary property (2.6) remains valid. Parallelly, the fuzzy labels $\alpha_{\mathbf{r},c_{\mathbf{v}_k}}$ can be optimized using $\frac{\partial lc\left(\mathbf{v}_k, c_{\mathbf{v}_k}\right)}{\partial \alpha_{\mathbf{r},c_{\mathbf{v}_k}}}$:

$$\triangle \alpha_{\mathbf{r},c_{\mathbf{v}_k}} = -u_\tau\left(\mathbf{r}|\mathbf{v}_k\right) \tag{2.14}$$

followed by subsequent normalization.

To adjust the window rule to now fuzzified values $\alpha_{\mathbf{r},c_{\mathbf{v}_k}}$ one considers T. Using the Gaussian form (2.3) for $u_\tau\left(\mathbf{r}|\mathbf{v}_k\right)$, the term T can be rewritten as

$$T = \left(\eta_{lc} - \eta_\alpha\right) \cdot \Pi\left(\alpha_{\mathbf{r},c_{\mathbf{v}_k}}\right)$$

with

$$\Pi\left(\alpha_{\mathbf{r},c_{\mathbf{v}_k}}\right) = \frac{\exp\left(-\frac{d(\mathbf{v}_k,\mathbf{w}_\mathbf{r})}{2\tau^2}\right)}{\sum_{\mathbf{r}'}\frac{\left(1-\alpha_{\mathbf{r},c_{\mathbf{v}_k}}-\alpha_{\mathbf{r}',c_{\mathbf{v}_k}}\right)}{\exp\left(\frac{d(\mathbf{v}_k,\mathbf{w}_{\mathbf{r}'})}{2\tau^2}\right)}} \tag{2.15}$$

and $\eta_\alpha = \alpha_{r,c_{v_k}}\left(1+\alpha_{r,c_{v_k}}\right)$ and η_{lc} in according to (2.10).

As in the original SNPC,

$$0 \leq lc\left(\mathbf{v}_k, c_{\mathbf{v}_k}\right)\left(1 - lc\left(\mathbf{v}_k, c_{\mathbf{v}_k}\right)\right) \leq 0.25$$

because $lc\left(\mathbf{v}_k, c_{\mathbf{v}_k}\right)$ fulfills the loss boundary property (2.6) [4]. Hence, one gets

$$-2 \leq T \leq 0.25$$

using the fact that $\alpha_{r,c_{v_k}} \leq 1$ [6]. Further, the absolute value of the factor T has to be significantly different from zero to have a valuable contribution in the update rule [4]. This yields the *window condition* $0 \ll |T|$, which can be obtained by balancing the local loss $lc\left(\mathbf{v}_k, c_{\mathbf{v}_k}\right)$ and the value of the assignment variable $\alpha_{r,c_{v_k}}$.

Subsequently the idea of metric adaptation is incorporated into FSNPC too [6],[19] now applying a *local* prototype dependent parametrized similarity measure $d\left(\mathbf{v}_k, \mathbf{w}_\mathbf{r}\right)$. Again, metric adaptation takes place as gradient descent on the cost function with respect to the relevance parameters $\lambda_\mathbf{r}$ (relevance learning):

$$\triangle\lambda_\mathbf{r} = -\frac{\partial lc\left(\mathbf{v}_k, c_{\mathbf{v}_k}\right)}{\partial\lambda_\mathbf{r}} \tag{2.16}$$

with

$$\frac{\partial lc\left(\mathbf{v}_k, c_{\mathbf{v}_k}\right)}{\partial\lambda_j\left(\mathbf{r}\right)} = -\frac{T}{2\tau^2}\cdot\frac{\partial d_\mathbf{r}^{\lambda_\mathbf{r}}\left(\mathbf{v}_k, \mathbf{w}_\mathbf{r}\right)}{\partial\lambda_j\left(\mathbf{r}\right)} \tag{2.17}$$

using the local cost (2.5) and subsequent normalization of the $\lambda_j\left(\mathbf{r}\right)$. In case of $\lambda = \lambda_\mathbf{r}$ for all $\mathbf{r}$ (global parametrized metric) one gets

$$\frac{\partial lc\left(\mathbf{v}_k, c_{\mathbf{v}_k}\right)}{\partial\lambda_j} = -\sum_\mathbf{r}\frac{T}{2\tau^2}\cdot\frac{\partial d^\lambda\left(\mathbf{v}_k, \mathbf{w}_\mathbf{r}\right)}{\partial\lambda_j} \tag{2.18}$$

In the following this variant is referred as FSNPC-R. In case of local relevance parameters the algorithm is denoted as FLSNPC-R. The computational complexity of the (F)SNPC methods can be estimated only roughly due to the nature of the stochastic gradient descent. To train an (L)(F)SNPC network for each cycle and for each datapoint of the training set $|\mathbf{W}|$ steps accounting for calculations related to prototype updates are needed. The number of cycles is typically related to the number of training samples, e.g. for 1000 samples 1000 training cycles maybe executed. For larger datasets (>> 1000 samples) in general only a random subset is selected and used for the optimization procedure. Especially the total number of sample queries used to train SNPC variants can be significantly reduced by use of active learning strategies as recently proposed in [20].

3 Supervised Neural GAS for fuzzy labeled data

Recently another fuzzified supervised LVQ algorithm has been proposed which is based on the well known Neural Gas algorithm as introduced in [21] and concepts taken from the

Supervised Relevance Neural GAS [17]. This new algorithm is known as Fuzzy Labeled Neural GAS (FLNG) [9] and will be reviewed in the following, compared with the above given FSNPC approach.

It differs from the above SNPC variants in such a way that the assumption of crisp classification for training data can be relaxed, i.e. a unique assignment of the data to the classes is no longer required. This is highly demanded in real world applications. For example, in medicine a clear (crisp) classification of data for training may be difficult or impossible: Assignments of a patient to a certain disorder frequently can be done only in a probabilistic (fuzzy) manner. Hence, it is of great interest to have a classifier which is able to manage this type of data.

We shortly review unsupervised Neural GAS and explain thereafter the supervised modification FLNG. We complete this part by transferring the ideas of relevance learning to FLNG too.

3.1 The neural gas network

Neural gas is an unsupervised prototype based vector quantization algorithm. It maps data vectors $\mathbf{v}$ from a (possibly high-dimensional) data manifold $V \subseteq \mathbb{R}^d$ onto a set A of neurons i formally written as $\Psi_{V \to A} : V \to A$. Thereby the notations as introduced in the section 2 are kept. Also in this case it is only supposed that the used distance measure $d(\mathbf{v}, \mathbf{w_i})$ is a differentiable symmetric similarity measure.

During the adaptation process a sequence of data points $\mathbf{v} \in V$ is presented to the map with respect to the data distribution $P(V)$. Each time the currently most proximate neuron s according to (2.1) is determined, and the pointer $\mathbf{w}_s$ as well as all pointers $\mathbf{w}_i$ of neurons in the neighborhood of $\mathbf{w}_s$ are shifted towards $\mathbf{v}$, according to

$$\triangle \mathbf{w}_i = -\epsilon h_\sigma (\mathbf{v}, \mathbf{W}, i) \frac{\partial d(\mathbf{v}, \mathbf{w}_i)}{\partial \mathbf{w}_i}. \tag{3.1}$$

The property of "being in the neighborhood of $\mathbf{w}_s$" is captured by the neighborhood function

$$h_\sigma (\mathbf{v}, \mathbf{W}, i) = \exp\left(-\frac{k_i (\mathbf{v}, \mathbf{W})}{\sigma}\right), \tag{3.2}$$

with the rank function

$$k_i (\mathbf{v}, \mathbf{W}) = \sum_j \theta \left(d(\mathbf{v}, \mathbf{w}_i) - d\left(\mathbf{v}, \mathbf{w}_j\right)\right) \tag{3.3}$$

counting the number of pointers $\mathbf{w}_j$ for which the relation $\|\mathbf{v} - \mathbf{w}_j\| < \|\mathbf{v} - \mathbf{w}_i\|$ is valid [21]. $\theta(x)$ is the Heaviside-function. It should be mentioned that the neighborhood function is evaluated in the input space. The adaptation rule for the weight vectors follows in average a potential dynamic according to the potential function [21]:

$$E_{NG} = \frac{1}{2C(\sigma)} \sum_j \int P(\mathbf{v}) h_\sigma (\mathbf{v}, \mathbf{W}, j) d\left(\mathbf{v}, \mathbf{w}_j\right) d\mathbf{v} \tag{3.4}$$

with $C(\sigma)$ being a constant. It will be dropped in the following. It was shown in many applications that the NG shows a robust behavior together with a high precision of learning.

3.2 Fuzzy Labeled NG

One can switch from the unsupervised scheme to a supervised scenario, i.e. each data vector is now accompanied by a label. According to the aim as explained above, the label is fuzzy: for each class k one has the possibilistic assignment $x_k \in [0,1]$ collected in the label vector $\mathbf{x} = (x_1, \ldots, x_{N_c})$. N_c is the number of possible classes. Further, fuzzy labels are introduced for each prototype $\mathbf{w}_j$: $\mathbf{y}_j = \left(y_1^j, \ldots, y_{N_c}^j\right)$. Now, the original unsupervised NG is adapted such that it is able to learn the fuzzy labels of the prototypes according to a supervised learning scheme. Thereby, the behavior of the original NG should be integrated as much as possible to transfer the excellent learning properties. This new algorithm is denoted as Fuzzy Labeled Neural Gas (FLNG). To include the fuzzy label accuracy into the cost function of FLNG a term to the usual NG cost function will be added, which judges the deviations of the prototype fuzzy labels from the fuzzy label of the data vectors:

$$E_{FLNG} = E_{NG} + \beta E_{FL} \tag{3.5}$$

The factor β is a balance factor, which could be under control or simply chosen as $\beta = 1$. For a precise definition of the new term E one has to differentiate between discrete and continuous data, which becomes clear during the derivation. The different situations are detailed in [9] and will not be reconsidered in the following. From the numerical analysis in [9] one can conclude that a Gaussian approach in modeling the rank replacement is suitable. Hence, only this specific variant of FLNG will be considered.

3.3 Gaussian kernel based FLNG

In the Gaussian approach, one weights the label error by a Gaussian kernel depending on the distance. Hence, the second term E_{FL} is chosen as

$$E_{FL} = \frac{1}{2} \sum_j \int P(\mathbf{v})\, g_\gamma\left(\mathbf{v}, \mathbf{w}_j\right) \left(\mathbf{x} - \mathbf{y}_j\right)^2 d\mathbf{v} \tag{3.6}$$

where $g_\gamma\left(\mathbf{v}, \mathbf{w}_j\right)$ is a Gaussian kernel describing a neighborhood range in the data space:

$$g_\gamma\left(\mathbf{v}, \mathbf{w}_j\right) = \exp\left(-\frac{d\left(\mathbf{v}, \mathbf{w}_j\right)}{2\gamma^2}\right) \tag{3.7}$$

Note that $g_\gamma\left(\mathbf{v}, \mathbf{w}_j\right)$ depends on the prototype locations, such that E_{FL} is influenced by both $\mathbf{w}$ and $\mathbf{y}$. Investigating this cost function, again, the first term $\frac{\partial E_{NG}}{\partial \mathbf{w}_i}$ of the full gradient $\frac{\partial E_{FLNG}}{\partial \mathbf{w}_i}$ is known from usual NG. The new second term now contributes according to

$$\frac{\partial E_{FL}}{\partial \mathbf{w}_i} = -\frac{1}{4\gamma^2} \int P(\mathbf{v})\, g_\gamma(\mathbf{v}, \mathbf{w}_i)\, \frac{\partial d(\mathbf{v}, \mathbf{w}_i)}{\partial \mathbf{w}_i} \left(\mathbf{x} - \mathbf{y}_i\right)^2 d\mathbf{v} \tag{3.8}$$

which takes the accuracy of fuzzy labeling into account for the weight update. Both terms define the learning rule for the weights.

For the fuzzy label one simply obtains $\frac{\partial E_{FLNG}}{\partial \mathbf{y}_i} = \frac{\partial E_{FL}}{\partial \mathbf{y}_i}$, where

$$\frac{\partial E_{FL}}{\partial \mathbf{y}_i} = -\int P(\mathbf{v})\, g_\gamma(\mathbf{v}, \mathbf{w}_i)\, (\mathbf{x} - \mathbf{y}_i)\, d\mathbf{v} \tag{3.9}$$

which is, in fact, a weighted average of the data fuzzy labels of those data belonging to the receptive field of the associated prototypes. However, in comparison to usual NG the

receptive fields are different because of the modified learning rule for the prototypes and their resulting different locations. The resulting learning rule is

$$\triangle \mathbf{y}_i = \epsilon_l g_\gamma \left(\mathbf{v}, \mathbf{w}_i\right) \left(\mathbf{x} - \mathbf{y}_i\right) \tag{3.10}$$

3.4 Relevance Learning for FLNG (FLNG-R)

In the theoretical derivation of the algorithm a general distance measure has been used, which can, in principle, be chosen arbitrarily, but sufficiently differentiable. Hence, a parametrized distance measure can be used as before in case of SNPC-R and FSNPC-R. For this purpose the derivatives are investigated

$$\frac{\partial E_{FLNG}}{\partial \lambda_k} = \frac{\partial E_{NG}}{\partial \lambda_k} + \beta \frac{\partial E_{FL}}{\partial \lambda_k} \tag{3.11}$$

One obtains:

$$\frac{\partial E_{NG}}{\partial \lambda_k} = \frac{1}{2C\left(\sigma\right)} \left(\begin{array}{c} \sum_j \int P\left(\mathbf{v}\right) h_\sigma \left(\mathbf{v}, \mathbf{W}, j\right) \frac{\partial d_\lambda \left(\mathbf{v}, \mathbf{w}_j\right)}{\partial \lambda_k} d\mathbf{v} \\ + \sum_j \int P\left(\mathbf{v}\right) d_\lambda \left(\mathbf{v}, \mathbf{w}_j\right) \frac{\partial h_\sigma \left(\mathbf{v}, \mathbf{W}, j\right)}{\partial \lambda_k} d\mathbf{v} \end{array} \right) \tag{3.12}$$

with $\frac{\partial h_\sigma (\mathbf{v}, \mathbf{W}, j)}{\partial \lambda_k} = -\frac{h_\sigma (\mathbf{v}, \mathbf{W}, j)}{\sigma} \cdot \frac{\partial k_j (\mathbf{v}, \mathbf{W})}{\partial \lambda_k}$. It is taken into account that the definition (3.3) of $k_j\left(\mathbf{v}, \mathbf{W}\right)$ with the derivative of the Heaviside-function $\theta\left(x\right)$ is the delta distribution $\delta\left(x\right)$. In this way one gets

$$\frac{\partial k_j\left(\mathbf{v}, \mathbf{W}\right)}{\partial \lambda_k} = \sum_l \delta\left(\triangle_\lambda\left(\mathbf{v}, \mathbf{w}_j, \mathbf{w}_l\right)\right) \cdot \frac{\partial \triangle_\lambda\left(\mathbf{v}, \mathbf{w}_j, \mathbf{w}_l\right)}{\partial \lambda_k} \tag{3.13}$$

with $\triangle_\lambda\left(\mathbf{v}, \mathbf{w}_j, \mathbf{w}_l\right) = d_\lambda\left(\mathbf{v}, \mathbf{w}_j\right) - d_\lambda\left(\mathbf{v}, \mathbf{w}_l\right)$. Hence in the second term (3.12) vanishes because δ is symmetric and non-vanishing only for $d_\lambda\left(\mathbf{v}, \mathbf{w}_j\right) = d_\lambda\left(\mathbf{v}, \mathbf{w}_l\right)$. Thus

$$\frac{\partial E_{NG}}{\partial \lambda_k} = \frac{1}{2C\left(\sigma\right)} \sum_j \int P\left(\mathbf{v}\right) h_\sigma\left(\mathbf{v}, \mathbf{W}, j\right) \frac{\partial d_\lambda\left(\mathbf{v}, \mathbf{w}_j\right)}{\partial \lambda_k} d\mathbf{v} \tag{3.14}$$

Now one pays attention to the second summand $\frac{\partial E_{FL}}{\partial \lambda_k}$ one has

$$\frac{\partial E_{FL}}{\partial \lambda_k} = -\frac{1}{4\gamma^2} \sum_j \int P\left(\mathbf{v}\right) g_\gamma\left(\mathbf{v}, \mathbf{w}_j\right) \frac{\partial d_\lambda\left(\mathbf{v}, \mathbf{w}_j\right)}{\partial \lambda_k} \left(\mathbf{x} - \mathbf{y}_j\right)^2 d\mathbf{v} \tag{3.15}$$

It should be mentioned that local relevance learning for FLNG-R can be introduced similar as within FSNPC-R but is not considered in the following. The computational complexity of the FLNG variants is mainly determined by the number of sample queries during the training of the networks. For each sample approximately $O(|\mathbf{W}| + |\mathbf{W}| \cdot log(|\mathbf{W}|))$ steps for prototype, metric and label calculations are needed. Thereby the term $|\mathbf{W}|$ refers to the typical calculation needed for each LVQ variant and the $log(|\mathbf{W}|)$ refers to the rank calculation which is a specific step for Neural GAS networks. The number of cycles is typically less or equal to the number of training samples. Again only a random subset query selection strategy may be applied for very large datasets ($>> 1000$) such that the number of queries can be limited by some prior knowledge about the data distribution.

4 Experiments and Applications

In the following experimental results for the application of the different developed variants of SNPC and Fuzzy Labeled Neural GAS are given. Thereby the SNPC results are compared with standard methods such as SNG and SVM, followed by a comparison of FSNPC with FLNG variants. Thereby, the usual Euclidean distance is applied. Further we investigate the behavior of the relevance learning variants using the scaled Euclidean metric (2.11). Then the parameter vector λ modifies the weighting of individual input dimensions with respect to the underlying optimization problem. Input dimensions with low relevance for the classification task are scaled which can be considered as a linear scaling of the input dimension restricted by a normalization constraint such that $\lambda_i \in [0, 1]$ with $i = 1, \ldots, D_v$. For $\lambda_i \approx 0$ the input dimensions are pruned in fact. This can be geometrically interpreted as a linear projection of the high dimensional data onto a lower dimensional data space. This choice allows a direct interpretation of the relevance parameters as a weighting of importance of the spectral bands for cancer detection, which may give a hint for potential biomarkers. In the analysis of the fuzzy algorithms we consider also the label error as a more specific indicator of the learning error which is defined as

$$\bar{y^2} = \frac{1}{|\mathcal{V}|} \sum_{r=1}^{|\mathbf{W}|} \sum_{i=1}^{|\Omega_\mathbf{r}|} \sum_{j=1}^{N_\mathcal{L}} (\mathbf{x_i^j} - \mathbf{y_r^j})^2 \quad \text{with } x_i \in \Omega_\mathbf{r} : \ i = 1, \ldots, |\Omega_\mathbf{r}|$$

This error measure is also given for some crisp calculation on the test sets. It should be noted that in the crisp case a miss classification counts simple as 2 giving label errors $\bar{y^2} \in [0.0, 2.0]$. For the fuzzy classification there is no such obvious relation between the classification and the label error because the classification error is obtained using a majority voting scheme and the labels can be arbitrary fuzzy.

4.1 Clinical data and experimental settings

The different clinical data sets used to show the capabilities of the algorithms are the Wisconsin Breast Cancer (WBC)[10], the leukemia data set (LEUK) provided by [11] and two other non-public Matrix Assisted Laser Desorption/Ionization mass spectrometry (MALDI-MS) proteomic data obtained from [12]. The WBC data set consists of 100 training samples and 469 test data, whereby for the training samples exactly half the data set is to cancer state. The spectra are given as 30-dimensional vectors. Detailed descriptions of the data including facts about preprocessing can be found in [10] for WBC. The LEUK data are obtained from plasma samples. A mass range between 1 to 10kDa was used. Details for the LEUK data can be found in [11].

The MALDI-MS data (PROT1, PROT2) are obtained by spectral analysis of serum of patients suffering from different cancer types and corresponding control probands. For the clinical preparations MB-HIC C8 Kits (Bruker Daltonik, Bremen, Germany) has been used. All purifications were performed in a one-step procedure according to the product description. Sample preparation onto the MALDI-TOF Anchor Chip target are done using alpha-cyano-4-hydroxy-cinnamic acid (HCCA) as matrix. Profiling spectra were generated on an autoflex MALDI-TOF MS (Bruker Daltonik, Bremen, Germany) in the linear mode for the PROT I data and on an UltraFlex MALDI-TOF MS (Bruker Daltonik, Bremen, Germany) for the PROT II data set. The obtained spectra were first processed using the standardized workflow as given in [22]. After preprocessing the LEUK spectra one obtains 145-dimensional vectors of peak areas. Thereby the LEUK data set consists of 74 cancer and 80 control samples. The PROT1 data set consists of 94 samples in two classes of nearly

	SNPC			SNG		SVM	
	train	test	y^2	train	test	train	test
WBC	98%	85%	0.3	67%	63%	97%	95%
LEUK	100%	100%	0.0	33%	30%	100%	96%
PROT1	95%	97%	0.06	52%	52%	100%	88%
PROT2	94%	80%	0.2	39%	37%	100%	82%

Table 1: Classification accuracy for the different cancer data sets for SNPC, SNG, SVM

	SNPC-R			LSNPC-R			SRNG		
	train	test	y^2	train	test	y^2	train	test	y^2
WBC	98%	94%	0.12	100%	96%	0.08	99%	94%	0.12
LEUK	100%	100%	0.0	100%	100%	0.0	100%	100%	0.0
PROT1	97%	91%	0.18	95%	76%	0.48	96%	90%	0.2
PROT2	95%	81%	0.38	96%	86%	0.28	82%	80%	0.4

Table 2: Classification accuracy for the different cancer data sets for SNPC-R, LSNPC-R, SRNG

equal size and 124 dimensions originating from the obtained peak areas. The PROT2 data are given by 203 samples in three classes with 78 dimensions.

For crisp classifications, 6 prototypes for WBC data and 2 prototypes for LEUK data were used. The PROT1 data set has been analyzed with 6 prototypes and the PROT2 data set using 9 prototypes, respectively. All training procedures has been done upto convergence with an upper limit of 5000 cycles. For the fuzzy variants of FLNG the number of prototypes has been changed in accordance to its data distribution dependent prototype learning property such that the LEUK and WBC model has been obtained using 6 prototypes, the PROT1 model using 12 prototypes and the PROT2 model using 15 prototypes.

The classification results for the standard crisp classification without metric adaptation are given in Tab. 1 and in Tab. 2 for crisp methods with metric adaptation. Clearly, metric adaptation significantly improves the classification accuracy. Some typical relevance profiles are depicted in Fig. 1. High relevance values refer to greater importance of the respective spectral bands for classification accuracy and, therefore, hints for potential biomarkers.

One can observe that SNPC-R is capable to generate suitable classification models typically leading to prediction rates above 91%. The results are in parts better than those obtained by ordinary SNPC. The results are reliable in comparison with SVM and SRNG. Besides the good prediction rates obtained from SNPC-R one gets additional information from the relevance profiles. For metrics per class one gets specific knowledge on important input dimensions per class.

Subsequently FSNPC and FLNG are considered with and without metric adaptation for the different data sets. As a first result from the simulations one can found that both algorithm need in general longer runtimes upto convergence, especially to sufficiently learn the underlying labeling. This can be explained due to the label learning of the prototypes, which not any longer is fixed from the startup such that the number of prototypes dedicated to represent a class can be determined during learning. The results depicted in Tab. 3 show reliable but a bit worse results with respect to the non fuzzy methods. FSNPC and FLNG behave similar but it should be mentioned that FSNPC is driven by a Gaussian mixture model approach whereas FLNG is motivated by statistical data clustering with

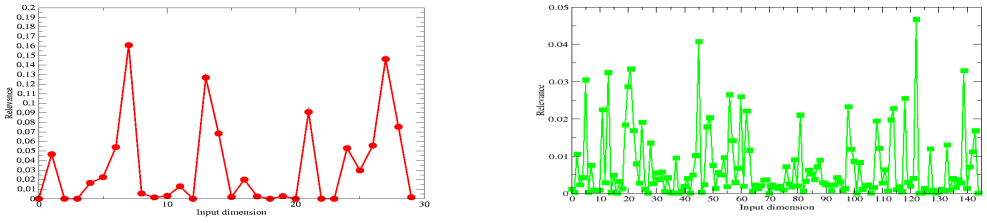


Figure 1: Relevance profiles for the WBC (left) and LEUK (right) data set using SNPC-R

	FSNPC				FLNG			
	train	y^2	test	y^2	train	y^2	test	y^2
WBC	99%	0.02	97%	0.06	88%	0.16	86%	0.18
LEUK	100%	0.0	93%	0.13	92%	0.11	79%	0.24
PROT1	98%	0.03	92%	0.16	83%	0.24	89%	0.18
PROT2	90%	0.17	70%	0.44	80%	0.28	78%	0.34

Table 3: Classification accuracy and label error for the labels ($\bar{y^2}$) for the different cancer data sets for FSNPC, FLNG

neighborhood cooperation.

Also for the fuzzy methods one can in general observe an improvement of the recognition and prediction accuracy by incorporating metric adaptation as depicted in Tab. 4. For the FLNG algorithm it could be observed that reliable models (measured on the recognition accuracy) needs typically twice as much prototypes as for FSNPC or other prototype based algorithms. This reflects, that the FLNG optimization is not just with respect to a given classification but also to the data distribution, which becomes a more critical factor for higher dimensional data.

For the fuzzy methods an additional measurement of convergence and accuracy, the label error (LE) becomes important. If the data could be sufficiently well represented by the prototype model the LE is a comparable measure for different models originating from

	FSNPC-R				FLSNPC-R				FLNG-R			
	train	y^2	test	y^2	train	y^2	test	y^2	train	y^2	test	y^2
WBC	98%	.03	99%	.02	99%	.03	99%	.02	91%	.13	92%	.14
LEUK	98%	.04	93%	.12	100%	.0	93%	.13	88%	.18	96%	.14
PROT1	98%	.03	97%	.05	97%	.06	94%	.1	83%	.22	79%	.21
PROT2	95%	.09	81%	.35	95%	.07	87%	.28	78%	.29	70%	.41

Table 4: Classification accuracies for cancer data sets using FSNPC-R, FLSNPC-R and FLNG-R. A classification of a data point is accounted for that class with the highest possibilistic value. The FSNPC derivatives behave similar to their crisp variants but a bit better than in comparison to FLNG. To obtain a reliable recognition accuracy for the LEUK, PROT1 and PROT2 data the number of prototypes had to be increased to $3, 6, 5$ per class. Label errors ($\bar{y^2}$) are given for the training and test data.

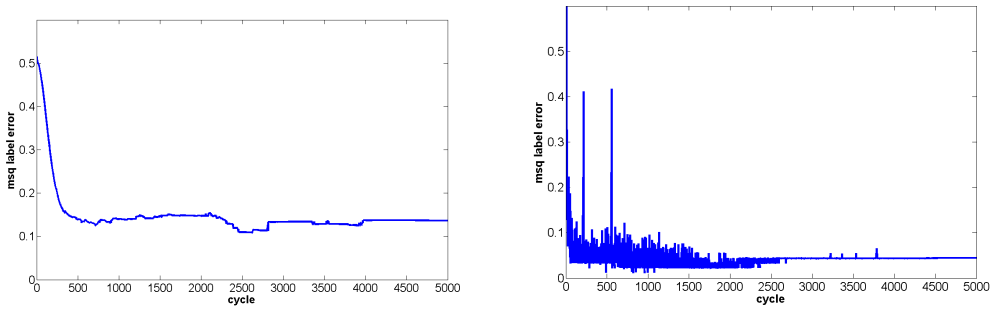


Figure 2: Typical convergence curve for label error (LE) using FLNG-R (left) and FSNPC-R (right) for the WBC data. To get a more stable analysis the algorithms has been trained fix with 5000 cycles to obtain these LE curves using 6 prototypes.

prototype fuzzy classifiers. An initial result is depicted in Figure 2 giving a first impression of LE behavior for the FLNG-R and FLSNPC-R algorithm. The LE in combination with the classification accuracy can be used as an indicator for the raw number of prototypes which should be used to get a sufficient modeling of the underlying data labeling and by considering this measure over time is a less raw measure for the current algorithm convergence than the pure accuracy, which typically is constant over large periods of learning. In Figure 2 one can see the LE's for FSNPC-R and FLNG-R in a comparison. Both algorithms show an overall convergence of the LE and end up with a similar error value. However for the FSNPC-R one finds a less stable behavior reflected by strong fluctuations in the middle of the learning task, which are vanishing in the convergence phase. For the FLNG-R changes in the LE are much smoother than for FSNPC-R. One can also observe that both algorithms get low LE's already at a very early cycle. Thereby the LE for FSNPC-R is finally a bit lower than for the FLNG-R algorithm within the different data sets. Considering the fuzzy labeling of the final prototype sets one can observe that both algorithms were capable to learn the labeling from the given training data. One finds prototypes with a very clear labeling, close to 100% for the corresponding class and hence a quite clear voronoi tessellation induced by this prototypes. But one can also find prototypes with lower safety in its class modeling and even prototypes, which show split decisions. Especially the last one are interesting in the sense that one immediately knows that decisions taken by those prototypes are doubtful and should be questioned.

5 Conclusion

The usual SNPC has been extended by relevance learning as one kind of metric adaptation and by fuzzy classification. A new adaptation dynamic for metric adaptation and prototype adjustment according to a gradient descent on a cost function has been derived. This cost function is obtained by appropriate modification of the SNPC. As demonstrated, this new soft nearest prototype classification with relevance learning can be efficiently applied to the classification of proteomic data and leads to results, which are competitive to results as reported by alternative state of the art algorithms. The extension of SNPC to fuzzy classification has been compared with the FLNG algorithm. The FSNPC algorithm with its motivation from Gaussian mixture approaches performed very well in the different experiments but contains some critical parameters such as the one in the window rule, which

may need to be adapted for some data by additional analysis. Also the estimations based on a Gaussian mixture approach may be inappropriate for non Gaussian data distributions. The FLNG in contrast strongly depends on the β control. In our analysis however it was observed that the proposed settings are in general well suited and the algorithms behave sufficiently stable with respect to these parametrization. It was found that the SNPC derivatives showed in parts better performance regarding classification. Using the label error as a more specific indicator of the learning behavior, the FSNPC algorithm shows a less stable learning behavior than FLNG, but better final LE values. This is probably referred to the specific learning dynamic of FSNPC, which is closely related to that of standard LVQ algorithms. The FLNG algorithm however does not any longer migrates the update behavior of LVQ algorithms and hence behaves different. This however brings the new possibility to allow learning of potentially fuzzy labeled data points, which was not possible in a direct way with prototype methods so far. From a practical point of view one can conclude that relevance learning in generally improves the classification accuracy of the algorithm and can be used to distinguish class specific input dimensions from less important features, which directly supports the search for biomarker candidates. Local relevance learning gives only small additional improvements for the prediction accuracy but can be useful to identify class specific properties of the data. Finally the fuzziness introduced in FSNPC and by FLNG gives the algorithm an additional freedom in determining the number of prototypes spend to a class. In case of FLNG one is now further able to support fuzzy labeled data as well, which allows the clinicians to keep the diagnosis fuzzy if necessary instead making it unnecessary strict. The presented prototype based classifiers are applicable also in non-clinical domains but they show some properties which make them very desirable in the context of clinical applications. The prototype approach generates simple easy interpretable models leading to group specific proteom profiles in case of proteomic data. The supported relevance learning allows a ranking of the importance of the individual input dimensions with respect to the classification task and can therefore be used to determine biomarker candidates. Also in the context of life long learning prototype based approach are well suited because they can be easily retrained if new (clinical) data become available. The new fuzzy properties are a further benefit for questions with unsafe labeled data or fuzzy decision processes as they often occur for clinical experiments.

ACKNOWLEDGMENT: The authors are grateful to E. Schaeffeler, U. Zanger, M. Schwab (all Dr. Margarete Fischer Institute für Klinische Pharmakologie Stuttgart, Germany), M. Stanulla, M. Schrappe (both Kinderklinik der Medizinischen Hochschule Hannover, Germany), T. Elssner and M. Kostrzewa (both Bruker Daltonik Leipzig, Germany) for providing the LEUK-dataset. The PROT1 and PROT2 data set has been provided by T. Elssner and M. Kostrzewa (both Bruker Daltonik Leipzig, Germany. The processing of the proteomic mass spectrometry data has been supported by the Bruker Daltonik GmbH using the CLINPROT^{TM} system.

References

[1] P. Binz, D. Hochstrasser, R. Appel, Mass spectrometry-based proteomics: current status and potential use in clinical chemistry, Clin. Chem. Lab. Med. 41 (12) (2003) 1540–1551.

[2] T. Kohonen, Self-Organizing Maps, Vol. 30 of Springer Series in Information Sciences, Springer, Berlin, Heidelberg, 1995, (2nd Ext. Ed. 1997).

[3] K. Crammer, R. Gilad-Bachrach, A.Navot, A.Tishby, Margin analysis of the lvq algorithm, in: Proc. NIPS 2002, 2002.

[4] S. Seo, M. Bode, K. Obermayer, Soft nearest prototype classification, IEEE Transaction on Neural Networks 14 (2003) 390–398.

[5] F.-M. Schleif, T. Villmann, B. Hammer, Local metric adaptation for soft nearest prototype classification to classify proteomic data, in: Fuzzy Logic and Applications: 6th Int. Workshop, WILF 2005, LNCS 2849/2006, Springer, 2006, pp. 290–296.

[6] T. Villmann, F.-M. Schleif, B. Hammer, Prototype-based fuzzy classification with local relevance for proteomics, Neurocomputing (2006) in press.

[7] T. Martinetz, S. Berkovich, K. Schulten, Neural-gas network for vector quantization and its application to time-series prediction, IEEE Transactions on Neural Networks 4 (4) (1993) 558–569.

[8] T. Villmann, B. Hammer, F.-M. Schleif, T. Geweniger, Fuzzy labeled neural gas for fuzzy classification, in: Proceedings of WSOM 2005, 2005, pp. 283–290.

[9] T. Villmann, B. Hammer, F.-M. Schleif, T. Geweniger, Fuzzy classification by fuzzy labeled neural gas, Neural Networks 19 (6-7) (2006) 772–779.

[10] C. Blake, C. Merz., UCI repository of machine learning databases., available at: http://www.ics.uci.edu/ mlearn/MLRepository.html (1998).

[11] M. Kostrzewa, Leukaemia study - internal results, Bruker Daltonik GmbH Bremen Department of Bioanalytics and MHH Hannover IKP Stuttgart (2004).

[12] M. Kostrzewa, Different proteomic cancer data, Bruker Daltonik GmbH Bremen (2005).

[13] B. Hammer, T. Villmann, Mathematical aspects of neural networks, in: M. Verleysen (Ed.), Proc. Of European Symposium on Artificial Neural Networks (ESANN'2003), d-side, Brussels, Belgium, 2003, pp. 59–72.

[14] T. Kohonen, S. Kaski, H. Lappalainen, Self-organized formation of various invariant-feature filters in the adaptive-subspace SOM, Neural Computation 9 (1997) 1321–1344.

[15] A. S. Sato, K. Yamada, Generalized learning vector quantization, in: G. Tesauro, D. Touretzky, T. Leen (Eds.), Advances in Neural Information Processing Systems, Vol. 7, MIT Press, 1995, pp. 423–429.

[16] B. Hammer, M. Strickert, T. Villmann, Supervised neural gas with general similarity measure, Neural Processing Letters 21 (1) (2005) 21–44.

[17] B. Hammer, T. Villmann, Generalized relevance learning vector quantization, Neural Networks 15 (8-9) (2002) 1059–1068.

[18] B.Hammer, F.-M.Schleif, T.Villmann, On the generalization ability of prototype-based classifiers with local relevance determination, Tech. Rep. Ifi-05-14, Clausthal University of Technology, Technical-Report, http://www.in.tu-clausthal.de/fileadmin/homes/techreports/ifi0514hammer.pdf (2005).

[19] T. Villmann, F.-M. Schleif, B. Hammer, Fuzzy labeled soft nearest neighbor classification with relevance learning, in: Proceedings of the International Conference of Machine Learning Applications (ICMLA'2005), IEEE Press, Los Angeles, 2005, pp. 11–15.

[20] F.-M. Schleif, B. Hammer, T. Villmann, Margin based active learning for LVQ networks, in: Proc. of 14th European Symposium on Artificial Neural Networks (ESANN) 2006, 2006, pp. 539–544.

[21] T. M. Martinetz, S. G. Berkovich, K. J. Schulten, 'Neural-gas' network for vector quantization and its application to time-series prediction, IEEE Trans. on Neural Networks 4 (4) (1993) 558–569.

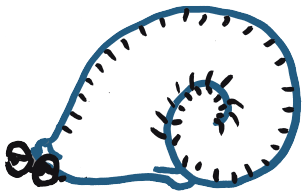
[22] B.-L. Adam, Y. Qu, J. Davis, M. Ward, M. Clements, L. Cazares, O. Semmes, P. Schellhammer, Y. Yasui, Z. Feng, G. Wright, Serum protein finger printing coupled with a pattern-matching algorithm distinguishes prostate cancer from benign prostate hyperplasia and healthy men, Cancer Research 62 (13) (2002) 3609–3614.

Chapter 3

Improved evaluation, interpretation and domain knowledge integration

3.1 Cancer Informatics by Prototype-networks in Mass Spectrometry

The article *Cancer Informatics by prototype networks in mass spectrometry* by F.-M.**Schleif**, T. Villmann, M. Kostrzewa, B. Hammer and A. Gammerman appeared in Artificial Intelligence in Medicine (45), p. 215-228, in 2009. In the article spectral data are processed as *functional data* by a wavelet based preprocessing and employing a functional metric. Prototype learners are extended by conformal prediction techniques used to obtain classification models discriminating the data and providing confidence and credibility measures for the decisions. The data were provided by M. Kostrzewa. I derived and implemented the algorithm, did all experiments and wrote the majority of the article. A. Gammerman provided support during the extension of prototype learners by conformal prediction techniques. T. Villmann and B. Hammer supervised the project. All authors discussed the general article.



Additional related publications where I am co-author are:

1. F.-M.**Schleif**, M. Lindemann, M. Diaz, P. Maa, J. Decker, T. Elssner, M. Kuhn, H. Thiele *Support Vector Classification of Proteomic Profile Spectra based on Feature Extraction with the Bi-orthogonal Discrete Wavelet Transform*, In *Computing and Visualization in Science* 12(4), p. 189–199, 2007. (Content: A wavelet based preprocessing and feature selection strategy for mass spectrometry data is proposed and evaluated on clinical proteomics data using a Support Vector Machine.)

Cancer Informatics by prototype networks in mass spectrometry

Frank-Michael Schleif[1]*, T. Villmann [1], M. Kostrzewa[2],
B. Hammer[3], and A. Gammerman[4]
[1]University Leipzig, Medical Department, Leipzig, Germany
[2]Bruker Daltonik GmbH, R & D, Leipzig, Germany
[3]Clausthal Univ. of Tech., Dept. of Math. and CS, Clausthal, Germany
[3]Royal Holloway University College London, London, UK

August 8, 2012

Abstract

Mass spectrometry has become a standard technique to analyse clinical samples in cancer research. The obtained spectrometric measurements reveal a lot of information of the clinical sample at the peptide and protein level. The spectra are high dimensional and, due to the small number of samples a sparse coverage of the population is very common. In clinical research the calculation and evaluation of classification models is important. For classical statistics this is achieved by hypothesis testing with respect to a chosen level of confidence. In clinical proteomics the application of statistical tests is limited due to the small number of samples and the high dimensionality of the data. Typically soft methods from the field of machine learning like prototype based vector quantizers [17], Support Vector Machines(SVM) [32], Self-Organizing Maps (SOMs) [17] and respective variants are used to generate such models. However for these methods the classification decision is crisp in general and no or only few additional information about the safety of the decision is available.

In this contribution the spectral data are processed as functional data by a wavelet based preprocessing [29] employing a functional metric [30, 28] in the prototype based classifiers. In particular, we demonstrate applications of the weighted Euclidean metric and the weighted functional norm (based on weighted L^p-norm) taking the specific nature of mass-spectra into account. This also allows the detection of potential biomarker candidates. To judge the classification decisions and model accuracy we focus on a method for the estimation of confidence using prototype based networks.

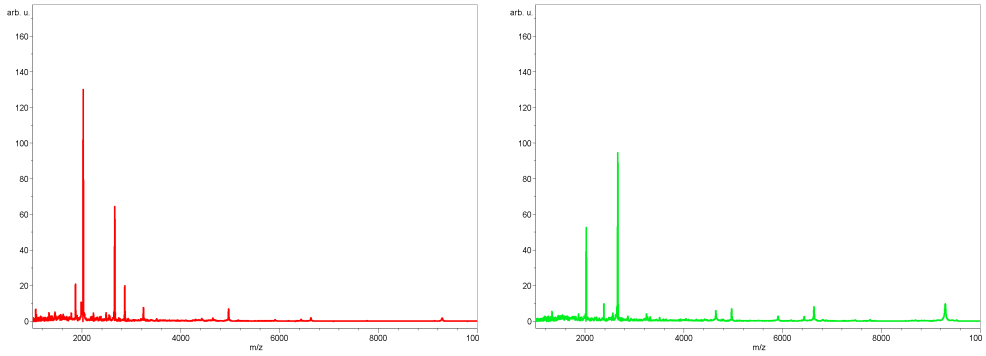
We demonstrate the usefulness of the above extensions in the analysis of mass spectra in proteomics and related knowledge discovery. In particular, we give application examples for biomarker detection based on feature selection and classification of spectra.

Keywords: clinical proteomics, cancer informatics, mass spectrometry, prototype classifiers confidence estimation

* *corresponding author, University Leipzig, Medical Department, email: schleif@informatik.uni-leipzig.de*

1 Introduction

Analysis of clinical proteomic spectra obtained from mass spectrometric measurements is a complicated issue [22]. One major objective is the search for potential biomarkers in complex body fluids like serum, plasma, urine, saliva, or cerebral spinal fluid [6, 24, 25, 10]. Typically the spectra are given as high-dimensional vectors. Thus, from a mathematical point of view, an efficient analysis and visualization of high-dimensional data sets is required. Moreover, the amount of available data is restricted: usually patient cohorts are small in comparison to the dimensionality of the data.



(a) Cancer Spectrum (b) Control Spectrum

Figure 1: (a) MALDI-TOF spectrum of a colorectal cancer patient and (b) a healthy subject after peptide isolation with C8 magnetic beads. On the Y-axis the relative intensity is shown. The mass to charge ratio (m/z) is demonstrated on the X-axis in Dalton. The spectra are already preprocessed (baseline correction,recalibration) using ClinProTools 2.1

In contrast to the widely applied multilayer perceptron [2], prototype based classification allows an easy interpretation, which is of particular interest for many (clinical) applications. One prominent prototype based classifier is the Supervised Relevance Neural Gas algorithm (SRNG)[12]. SRNG leads to a robust classifier where efficient learning of labeled high dimensional data is possible and has been already used in different types experiments [37, 27, 38, 34].

In general the available approaches to model classifiers in clinical proteomics initially transform the spectra into a vector space followed by training a classifier. In this way the functional nature of the data is lost, which may lead to suboptimal classifier models. A functional representation of the data with respect to the used metric and a weighting or pruning of (priorly not known) irrelevant parts of the inputs, would be desirable. A discriminative data representation is necessary. The extraction of such discriminant features is difficult for spectral data and typically done by a parametric peak picking procedure. This peak picking is often the focus of criticism because some present peaks may not be detected and the functional nature of the data is partially lost. To avoid this difficulties we focus on the approach as given in [30, 28] and apply a wavelet encoding to the spectral data to get discriminative features. The obtained wavelet coefficients are sufficient to reconstruct the signal, still containing all relevant information of the spectra in a *functional* encoding. However this better discriminating set of features is typically more complex and hence a robust approach to determine the desired classification model is needed. Taking this into account a feature selection is applied based on a statistical pre-analysis of the data and the

SRNG algorithm is used to obtained predictive models.

In this contribution, we focus on the conformal prediction concept incorporated in prototype based learning vector quantizers (LVQ). The paper is organized as follows. First we briefly review the functional encoding of mass spectrometric data by means of a wavelet based encoding. Subsequently the theory of the Supervised Relevance Neural Gas (SRNG) and its equipment with a functional metric is reviewed. After these settings, the method of conformal prediction [39, 9] is reviewed and we show how it can be used together with LVQ approaches. Subsequently the methodology is applied on experimental data from two clinical proteom studies. We evaluate the results not only using cross validation but also in the light of conformal prediction which allows the assessment of the classification safety by means of p-values as known from classical statistics.

2 Preprocessing

The classification of mass spectra involves multiple preprocessing steps. In general peak picking is used to locate and quantify positions of peaks within the spectrum and feature extraction is applied on the peak list to obtained an adequate feature matrix. In the first step a number of procedures as baseline correction, optional denoising, noise estimation and normalization are needed[16, 26]. Upon these prepared spectra the peaks have to be identified by scanning all local maxima and the associated peak endpoints followed by a S/N thresholding such that one obtains the desired peak list.

The procedure of baseline correction and recalibration (alignment) of multiple spectra is standard, and has been done using ClinProTools (details in [16])[1]. Here we propose an alternative feature extraction procedure preserving all (potentially small) peaks containing relevant information by use of the discrete wavelet transformation (DWT). The feature extraction has been done by Wavelet analysis using the Matlab Wavelet-Toolbox[2], due to the local analysis property of wavelet analysis the features can still be related back to original mass position in the spectral data which is essential for further biomarker analysis. In a first step a feature selection procedure using the Kolmogorov-Smirnoff test (KS-test) was applied. The test was used to identify features which show a significant ($p < 0.01$) discrimination between the two groups (cancer,control). To get valid results a p-value adjustment by means of the bonferroni-correction has been applied as well. This is done in accordance to [40] where also a generation to a multiclass experiment is given.

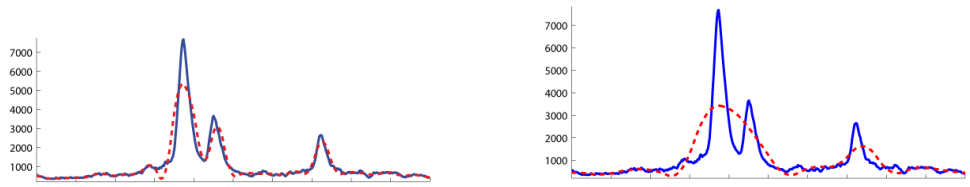
2.1 Feature Extraction by Bi-orthogonal Discrete Wavelet Transform

Wavelets have been developed as powerful tools [1, 19] used for noise removal and data compression. The discrete version of the continuous wavelet transform leads to the concept of a multiresolution analysis (MRA). This allows a fast and stable wavelet analysis and synthesis. The analysis becomes more precise if the wavelet shape is adapted to the signal to be analyzed. For this reason one can apply the so called bi-orthogonal wavelet transform[3] which uses two pairs of scaling and wavelet functions. One is for the decomposition/analysis and the other one for reconstruction/synthesis. The advantage of the bi-orthogonal wavelet transform is the higher degree of freedom for the shape of the scaling and wavelet function.

In our analysis such a smooth synthesis pair was chosen to avoid artifacts. It can be expected that a signal in the time domain can be represented by a small number of a

[1]Biomarker software available at http://www.bdal.de

[2]The Matlab Wavelet-Toolbox can be obtained from www.mathworks.com



(a) Wavelet reconstruction $L = 4$ (b) Wavelet reconstruction$L = 5$

Figure 2: Wavelet reconstruction of the spectra with $L = 4, 5$, x measurement positions, y-arbitrary unit. The original signal is plotted with the interrupted line (blue) and the reconstruction with the solid with a white band inside. One observes that a wavelet analysis with $L = 5$ is too rough to approximate the sharp peaks.

relatively large set of coefficients from the wavelet domain. The spectra are reconstructed in dependence of a certain approximation level L of the MRA which can be considered as a hard-thresholding. The denoised spectrum looks similar to the reconstruction as depicted in Figure 2. The starting point for an argumentation is the simplest example of a MRA which can be defined by the characteristic function $\chi_{[0,1)}$. The corresponding wavelet is the so-called *Haar* wavelet. Assume that the denoised spectrum $f \in L_2(\mathbb{R})$ has a peak with endpoints $2^j k$ and $2^j(k+1)$, the integral of the peak can be written as

$$\int_{2^j k}^{2^j(k+1)} f(t)dt = \int_{\mathbb{R}} f(t)\chi_{[2^j k, 2^j(k+1))}(t)dt$$

Obviously the right hand side is the Haar DWT scaling coefficient $c_{j,k} = \langle f, \psi_{j,k} \rangle$ at scale $a = 2^j$ and translation $b = 2^j k$.

One obtains approximation- and detail-coefficients [3]. The approximation coefficients describe a generalized peak list of the denoised spectrum encoding primal spectral information and depend on the level L which is determined with respect to the measurement procedure. For linearly MALDI-TOF spectra a device resolution of $500 - 800Da$ can be expected. This implies limits to the minimal peak width in the spectrum and hence, the reconstruction level of the Wavelet-Analysis should be able to model corresponding peaks. A level $L = 4$ is typically sufficient for a linear measured spectrum with ≈ 20000 measurement points (see Figure 2). The level L can be automatically determined by considering expected peak width in Da and the reconstruction capabilities of wavelet analysis at a given level. Alternatively multiple levels can be tried and a standard peak picking approach can be applied on both, the original and the reconstructed spectrum. If the obtained peak lists are sufficiently similar, which means, that at least peaks with good S/N values in the original spectrum are sufficiently recovered in the reconstruction the taken level can be considered as acceptable for the experiment.

Applying this procedure including the KS-test on the spectra with an initial number of $\approx$ 4000 measurement points in a range of $1500 - 3500Da$ per spectrum one obtains 416 wavelet coefficients used as representative features per spectrum, still allowing a reliable functional representation of the data. An application of the KS-Test still keeps 101 coefficients for the final analysis of the colorectal cancer patients (CRC) data set and 40 coefficients for the lung cancer (LC) data set[3].

[3]The ks-test is an optional data reduction step, the removed dimensions are in general neighbored, closed stripes of noise and not discriminating signals

3 Bioinformatic methods

The Supervised Relevance Neural Gas (SRNG) algorithm is a prototype based classification model, which will be introduced very briefly. Subsequently we extend the concept of conformal prediction as introduced in [39, 9] in the context of prototype based networks which is used in the evaluation part to determine confidence values for obtained classification results.

3.1 Supervised Relevance Neural Gas with generalized metrics

Supervised Neural Gas (SNG) is considered as a representative for prototype based classification approaches as introduced by KOHONEN. Different prototype classifiers have been proposed so far [17, 23, 14, 36] as improvements of the original approach. The SNG has been introduced in [36] and combines ideas from the Neural Gas algorithm (NG) introduced in [20] with the Generalized learning vector quantizer (GLVQ) as given in [23]. Subsequently we give the basic notations and some remarks to the integration of alternative metrics into Supervised Neural Gas (SNG). Details on SNG including convergence proofs can be found in [36].

Let us first clarify some notations: Let $c_{\mathbf{v}} \in \mathcal{L}$ be the label of input $\mathbf{v}$, $\mathcal{L}$ a set of labels (classes) with $\#\mathcal{L} = N_{\mathcal{L}}$. Let $V \subseteq \mathbb{R}^{D_V}$ be a finite set of inputs $\mathbf{v}$. LVQ uses a fixed number of prototypes (weight vectors, codebook vectors) for each class. Let $\mathbf{W} = \{\mathbf{w}_{\mathbf{r}}\}$ be the set of all codebook vectors and $c_{\mathbf{r}}$ be the class label of $\mathbf{w}_{\mathbf{r}}$. Furthermore, let $\mathbf{W}_c = \{\mathbf{w}_{\mathbf{r}} | c_{\mathbf{r}} = c\}$ be the subset of prototypes assigned to class $c \in \mathcal{L}$.

The task of vector quantization is realized by the map Ψ as a winner-take-all rule, i.e. a stimulus vector $\mathbf{v} \in V$ is mapped onto the closest $\mathbf{w}_s$,

$$\Psi^{\lambda}_{V \to \mathcal{A}} : \mathbf{v} \mapsto \mathbf{s}(\mathbf{v}) = \mathrm{argmin}_{\mathbf{r} \in A} d^{\lambda}(\mathbf{v}, \mathbf{w}_{\mathbf{r}}) \tag{1}$$

with $d^{\lambda}(\mathbf{v}, \mathbf{w})$ being an arbitrary differentiable distance measure[4] which may depend on a parameter vector λ and A a (ordered) grid of neurons. Subsequently we only expect that the used distance measure is differentiable with respect to its parameters. For the moment we take λ as fixed. The neuron $\mathbf{s}(\mathbf{v})$ is called winner or best matching unit. The subset of the input space

$$\Omega^{\lambda}_{\mathbf{r}} = \{\mathbf{v} \in V : \mathbf{r} = \Psi_{V \to A}(\mathbf{v})\} \tag{2}$$

which is mapped to a particular neuron $\mathbf{r}$ according to (1), forms the (masked) receptive field of that neuron forming a Voronoi tessellation. If the class information of the weight vector is used, the boundaries $\partial\Omega^{\lambda}_{\mathbf{r}}$ generate the decision boundaries for classes. A training algorithm should adapt the prototypes such that for each class $c \in \mathcal{L}$, the corresponding codebook vectors $\mathbf{W}_c$ represent the class as accurately as possible. This means that the set of points in any given class $V_c = \{\mathbf{v} \in V | c_{\mathbf{v}} = c\}$, and the union $\mathcal{U}_c = \bigcup_{\mathbf{r}|_{\mathbf{w}_{\mathbf{r}} \in \mathbf{W}_c}} \Omega_{\mathbf{r}}$ of receptive fields of the corresponding prototypes should differ as little as possible.

Supervised Neural Gas (SNG) constitutes a method to train prototypes efficiently according to given data points. Again, let $\mathbf{W}_c = \{\mathbf{w}_{\mathbf{r}} | c_{\mathbf{r}} = c\}$ be the subset of prototypes assigned to class $c \in \mathcal{L}$ and K_c its cardinality.

Further we assume to have m data vectors $\mathbf{v}_i$. As pointed out in [36], neighborhood learning for a given input $\mathbf{v}_i$ with label c is applied to the subset $\mathbf{W}_c$. The respective cost

[4] A distance measure is a non-negative real-valued function, which, in contrast to a metric does not necessarily fulfill the triangle inequality and the symmetry property. For prototype algorithms of the mentioned type the used distance measure need not to be a metric. A detailed discussion of this fact with respect to the considered methods is available in [11, 13]

function is

$$Cost_{SNG}(\gamma) = \sum_{i=1}^{m} \sum_{\mathbf{r}|\mathbf{w}_\mathbf{r} \in \mathbf{W}_{c_i}} \frac{h_\gamma(\mathbf{r}, \mathbf{v}_i, \mathbf{W}_{c_i}) \cdot f(\mu_\lambda(\mathbf{r}, \mathbf{v}))}{C(\gamma, K_{c_i})} \tag{3}$$

with $f(x) = (1 + \exp(-x))^{-1}$, $h_\gamma(\mathbf{r}, \mathbf{v}, \mathbf{W}) = \exp\left(-\frac{k_\mathbf{r}(\mathbf{v}, \mathbf{W})}{\gamma}\right)$ and $\mu_\lambda(\mathbf{r}, \mathbf{v}) = \frac{d^\lambda_\mathbf{r} - d^\lambda_{\mathbf{r}_-}}{d^\lambda_\mathbf{r} + d^\lambda_{\mathbf{r}_-}}$ and $d^\lambda_{\mathbf{r}_-}$ is defined as the squared distance to the best matching prototype but labeled with $c_{\mathbf{r}_-} \neq c_\mathbf{v}$, say $\mathbf{w}_{\mathbf{r}_-}$ and $d^\lambda_\mathbf{r} = d^\lambda(\mathbf{v}, \mathbf{w}_\mathbf{r})$. Details on the corresponding update rules are given in [36].

3.1.1 Incorporation of a functional metric to SNG

As pointed out before, the distance measure $d^\lambda(\mathbf{v}, \mathbf{w})$ is only required to be differentiable with respect to λ and $\mathbf{w}$. The triangle inequality has not to be fulfilled necessarily. This leads to a great freedom in the choice of suitable measures and allows the usage of non-standard metrics in a natural way. We now review the functional metric as given in [18], the obtained derivations can be plugged into the above equations leading to SNG with a functional metric, the data are functions represented by vectors and, hence, the vector dimensions are spatially correlated.

Common vector processing does not take the spatial order of the coordinates into account. As a consequence, the functional aspect of spectral data is lost. For proteom spectra the order of signal features (peaks) is due to the nature of the underlying biological samples and the measurement procedure. The masses of measured chemical compounds are given ascending and peaks encoding chemical structures with a higher mass follow chemical structures with lower masses. In addition multiple peaks with different masses may encode parts of the same chemical structure and hence are correlated.

In [18] a distance measure has been proposed taking the functional structure of the data into account, involving the previous and next values of v_i in the i-th term of the sum, instead of v_i alone. V can be represented as $V = (v_1, \ldots, v_D)$. Assuming a constant sampling period τ, the proposed norm is:

$$\mathcal{L}^{fc}_p(\mathbf{v}) = \left(\sum_{k=1}^{D} (A_k(\mathbf{v}) + B_k(\mathbf{v}))^p\right)^{\frac{1}{p}} \tag{4}$$

with

$$A_k(\mathbf{v}) = \begin{cases} \frac{\tau}{2}|v_k| & \text{if } 0 \leq v_k v_{k-1} \\ \frac{\tau}{2}\frac{v_k^2}{|v_k| + |v_{k-1}|} & \text{if } 0 > v_k v_{k-1} \end{cases} \tag{5}$$

$$B_k(\mathbf{v}) = \begin{cases} \frac{\tau}{2}|v_k| & \text{if } 0 \leq v_k v_{k+1} \\ \frac{\tau}{2}\frac{v_k^2}{|v_k| + |v_{k+1}|} & \text{if } 0 > v_k v_{k+1} \end{cases} \tag{6}$$

are respectively the triangles on the left and right hand sides v_i. Just as for L_p, the value of p is assumed to be a positive integer. At the left and right ends of the sequence, v_0 and v_D are assumed to be equal to zero. The derivatives for the functional metric taking $p = 2$ are given in [18].

Now we consider the scaled functional norm where each dimension v_i is scaled by a parameter $\lambda_i \geq 0$ $\lambda_i \in (0, 1]$ and $\sum_i \lambda_i = 1$. Then the scaled functional norm is:

$$\mathcal{L}^{fc}_p(\lambda\mathbf{v}) = \left(\sum_{k=1}^{D} (A_k(\lambda\mathbf{v}) + B_k(\lambda\mathbf{v}))^p\right)^{\frac{1}{p}} \tag{7}$$

with

$$A_k(\lambda\mathbf{v}) = \begin{cases} \frac{\tau}{2}\lambda_k |v_k| & \text{if } 0 \le v_k v_{k-1} \\ \frac{\tau}{2}\frac{\lambda_k^2 v_k^2}{\lambda_k|v_k|+\lambda_{k-1}|v_{k-1}|} & \text{else} \end{cases} \qquad B_k(\lambda\mathbf{v}) = \begin{cases} \frac{\tau}{2}\lambda_k |v_k| & \text{if } 0 \le v_k v_{k+1} \\ \frac{\tau}{2}\frac{\lambda_k^2 v_k^2}{\lambda_k|v_k|+\lambda_{k+1}|v_{k+1}|} & \text{else} \end{cases} \tag{8}$$

The prototype update changes to:

$$\frac{\partial \delta_2^2(\mathbf{x},\mathbf{y},\lambda)}{\partial x_k} = \frac{\tau^2}{2}(2 - U_{k-1} - U_{k+1})(V_{k-1} + V_{k+1})\triangle_k \tag{9}$$

with

$$U_{k-1} = \begin{cases} 0 \text{ if } 0 \le \triangle_k \triangle_{k-1} \\ \left(\frac{\lambda_{k-1}\triangle_{k-1}}{\lambda_k|\triangle_k|+\lambda_{k-1}|\triangle_{k-1}|}\right)^2 \text{ else} \end{cases} \quad , \quad U_{k+1} = \begin{cases} 0 \text{ if } 0 \le \triangle_k \triangle_{k+1} \\ \left(\frac{\lambda_{k+1}\triangle_{k+1}}{\lambda_k|\triangle_k|+\lambda_{k+1}|\triangle_{k+1}|}\right)^2 \text{ else} \end{cases}$$

$$V_{k-1} = \begin{cases} 1\lambda_k \text{ if } 0 \le \triangle_k \triangle_{k-1} \\ \frac{\lambda_k|\triangle_k|}{\lambda_k|\triangle_k|+\lambda_{k-1}|\triangle_{k-1}|} \text{ else} \end{cases} \quad , \quad V_{k+1} = \begin{cases} 1\lambda_k \text{ if } 0 \le \triangle_k \triangle_{k+1} \\ \frac{\lambda_k|\triangle_k|}{\lambda_k|\triangle_k|+\lambda_{k+1}|\triangle_{k+1}|} \text{ else} \end{cases}$$

and $\triangle_k = x_k - y_k$ For the λ-update one observes:

$$\begin{aligned}
\frac{\partial \mathcal{L}_p^{fc}(\lambda\mathbf{v})}{\partial \lambda_k} &= \frac{\partial \left(\sum_{k=1}^{D}(A_k(\lambda\mathbf{v}) + B_k(\lambda\mathbf{v}))^p\right)^{\frac{1}{p}}}{\partial \lambda_k} \\
&= p\left(\sum_{k=1}^{D}\left(A_{k-1}(\lambda\mathbf{v}) + A_{k+1}(\lambda\mathbf{v})\right)^p\right)^{\frac{1-p}{p}} \frac{\partial\left[\sum_{k=1}^{D}(A_k(\lambda\mathbf{v}) + B_k(\lambda\mathbf{v}))^p\right]}{\partial \lambda_k} \\
&= C_p \frac{\partial\left[\sum_{k=1}^{D}(A_k(\lambda\mathbf{v}) + B_k(\lambda\mathbf{v}))^p\right]}{\partial \lambda_k} \\
&= C_p \frac{\sum_{k=1}^{D}\partial\left[(A_k(\lambda\mathbf{v}) + B_k(\lambda\mathbf{v}))^p\right]}{\partial \lambda_k} \\
&= C_p \frac{\partial\left[\left(A_{k-1}(\lambda\mathbf{v}) + B_{k-1}(\lambda\mathbf{v})\right)^p + (A_k(\lambda\mathbf{v}) + B_k(\lambda\mathbf{v}))^p + \left(A_{k+1}(\lambda\mathbf{v}) + B_{k+1}(\lambda\mathbf{v})\right)^p\right]}{\partial \lambda_k} \\
&= C_p\left(c_p^{k-1}\frac{\partial\left[A_{k-1}(\lambda\mathbf{v}) + B_{k-1}(\lambda\mathbf{v})\right]}{\partial \lambda_k} + c_p^k \frac{\partial\left[A_k(\lambda\mathbf{v}) + B_k(\lambda\mathbf{v})\right]}{\partial \lambda_k} + c_p^{k+1}\frac{\partial\left[A_{k+1}(\lambda\mathbf{v}) + B_{k+1}(\lambda\mathbf{v})\right]}{\partial \lambda_k}\right)
\end{aligned}$$

with the following expressions

$$\begin{aligned}
c_p^j &= p \cdot (A_j(\lambda\mathbf{v}) + B_j(\lambda\mathbf{v}))^{p-1} \\
&= p \cdot \left(\begin{cases} \frac{\tau}{2}\lambda_j |v_j| & \text{if } 0 \le v_j v_{j-1} \\ \frac{\tau}{2}\frac{\lambda_j^2 v_j^2}{\lambda_j|v_j|+\lambda_{j-1}|v_{j-1}|} & \text{if } 0 > v_j v_{j-1} \end{cases} + \begin{cases} \frac{\tau}{2}\lambda_j |v_j| & \text{if } 0 \le v_j v_{j+1} \\ \frac{\tau}{2}\frac{\lambda_j^2 v_j^2}{\lambda_j|v_j|+\lambda_{j+1}|v_{j+1}|} & \text{if } 0 > v_j v_{j+1} \end{cases} \right)^{p-1}
\end{aligned}$$

putting all together and with some minor mathematical transformations one obtains:

$$\begin{aligned}
\frac{\partial \mathcal{L}_p^{fc}(\lambda\mathbf{v})}{\partial \lambda_k} &= C_p \begin{cases} 0 + c_p^k\left(\frac{\tau}{2}|v_k|\right) & \text{if } 0 \le v_{k-1}v_k \\ \frac{1}{2}\tau\frac{\lambda_k^2 c_p^k v_k^2|v_k| - c_p^{k-1}|v_k|v_{k-1}^2\lambda_{k-1}^2 + 2\lambda_k c_p^k v_k^2|v_{k-1}|\lambda_{k-1}}{\left(\lambda_k|v_k| + |v_{k-1}|\lambda_{k-1}\right)^2} & \text{if } 0 > v_{k-1}v_k \end{cases} \\
&\quad + C_p \begin{cases} c_p^k\left(\frac{\tau}{2}|v_k|\right) + 0 & \text{if } 0 \le v_{k+1}v_k \\ \frac{1}{2}\tau\frac{\lambda_k^2 c_p^k v_k^2|v_k| - c_p^{k+1}|v_k|v_{k+1}^2\lambda_{k+1}^2 + 2\lambda_k c_p^k v_k^2|v_{k+1}|\lambda_{k+1}}{\left(\lambda_k|v_k| + |v_{k+1}|\lambda_{k+1}\right)^2} & \text{if } 0 > v_{k+1}v_k \end{cases}
\end{aligned}$$

Using this parametrization one can emphasize/neglect different parts of the function for classification. This distance measure can be put into SNG as shown above and has been applied subsequently in the analysis of clinical proteom spectra. SNG with metric adaptation is subsequently referred as SRNG.

4 Evaluation of Prototype based classifier models

Advanced prototype based classification models show typically high regularisation capabilities [11]. Nevertheless also the results of prototype networks need a thoroughly analysis by cross validation to get practical measures to rate the prediction capabilities of the current model. Beside these generic measures of confidence in the results obtained by a classification model a more fine grained confidence analysis would be desirable. Classical statistics typically allows a judgment on the classification accuracy of a single item by means of p-values [15] but are not applicable (in a valid sense) for these type of data, in general. Also Gaussian Mixture Models allow to determine the probability of a classification decision, but make additional constraints on the considered type of data [15]. This techniques are well understood but in general not available for soft methods like SVM or prototype networks. Only few attempts were made to give reliability estimate for these soft methods (see e.g. [4, 5]). Thereby the reliability estimate can be helpful to judge on the reliability of a decision but also in a more generic framework to improve the overall performance of the classifier. Reliability sometimes also referred as confidence, has been subject of a quite new theory called conformal prediction as introduced in [39] which fills this gap under some moderate constraints. Here we show how the concept of conformal prediction can be applied to prototype networks and allows the determination of statistical significance values as needed in clinical studies and cancer informatics.

4.1 Conformal Prediction for Prototype based Networks

Conformal predictors aim at the estimation of confidence of a given classification decision. They remain automatically valid (in average) under the randomness assumption [39, 9]. It is assumed, that the objects and their labels are generated independently from the same probability distribution. This appears to be a strong assumption but in fact it is a much weaker assumption than assuming a parametric statistical model. Conformal predictors never overrate the accuracy and reliability of their predictions [39, 9]. When the stochastic mechanism significantly deviates from the model, conformal predictors remain valid but their efficiency inevitably suffers [39, 9]. As conformal predictors are provably valid, efficiency with respect to computational performance as well as with respect to the effort to extend a classifier to a conformal predictor, are the only things which we need to worry about. First we will give some basic notations and review the main concepts of conformal prediction as given in [39, 9].

4.1.1 Conformal prediction a brief overview

We now briefly review the concepts of conformal prediction as presented [7] and the tutorial given in [31]. The basics of conformal prediction rely on confidence intervals from classical statistics and are well theoretically founded [8]. Here we focus on classification and deal with labeled data. The task is: predict each label after seeing its object:

- from x_1 predict y_1
- from $(x_1, y_1), x_2$, predict y_2
- from $(x_1, y_1), (x_2, y_2), x_3$ predict y_3 and so on

Here we assume *randomness*. In reality we choose the examples independently from some probability distribution $\mathbb{Q}$ on $\mathbb{Z} = \mathbb{D} \times \mathbb{Y}$. The samples are independent and identically distributed. And we do not make any assumptions about $\mathbb{Q}$. Usually independence can be weakened to exchangeability [31]. To do the prediction with confidence we write $\mathbb{Z}^*$ for the set of all finite sequences of elements of $\mathbb{Z}$ such that:

$$\mathbb{Z}^* = \cup_{n=0}^{\infty} Z^n$$

A level $(1-\epsilon)$ confidence predictor is a mapping

$$\Gamma : \mathbb{Z}^* \times \mathbb{D} \to 2^{\mathbb{Y}}$$

after observing old examples $z_1, \ldots, z_{n-1}$ and the new object x_n, we predict that the label of (x_n, y_n) will be in the subset

$$\Gamma(z_1, \ldots, z_{n-1}, x_n)$$

of the label space Y. A $(1-\epsilon)$ confidence predictor is exactly valid if its hits are independent and all happen with probability $(1-\epsilon)$. It is conservatively valid if the probability that the predictions on rounds $n_1, \ldots, n_k$ are all hits is always at least $(1-\epsilon)^k$. This does not depend on a specific probability function. Valid confidence predictors are constructed from nonconformity measures by means of real values functions A: $(x_1, y_1), \ldots, (x_{n-1}, y_{n-1}), (x, y)$ as a measure of how different (x, y) is from $(x_1, y_1), \ldots, (x_{n-1}, y_{n-1})$. Here one predicts values of y_n that make x_n, y_n differ minimally from the rest. From a given nonconformity measure, we construct a $(1-\epsilon)$ confidence predictor Γ^ϵ for every $\epsilon \in [0, 1]$, and they are nested in the natural way: $\Gamma^{\epsilon_1}(z_1, \ldots, z_n, x_n) \subseteq \Gamma^{\epsilon_2}(z_1, \ldots, z_n, x_n)$ when $\epsilon_1 \geq \epsilon_2$. The more confident one wants to be, the larger the region must be chosen. So e.g. $\Gamma^{0.05}$ the prediction region, is a set that contains the true labeling with a probability of at least 95%. Typically $\Gamma^{0.05}$ also contains the prediction $\hat{y}$. We call $\hat{y}$ the point prediction. In case of classification $\Gamma^{0.05}$ may consist of a few of these values or, in the best case, just one [31]. Given a nonconformity measure, the conformal prediction algorithm produces a prediction region Γ^ϵ for every probability of ϵ. The region for Γ^ϵ is a $1-\epsilon$ prediction interval which contains $\hat{y}$. with a probability of at least $1-\epsilon$. The regions for different ϵ are nested: when $\epsilon_1 \leq \epsilon_2$: so that $1-\epsilon_1$ is a lower level of confidence than $1-\epsilon_2$, we have $\Gamma^{\epsilon_1} \subset \Gamma^{\epsilon_2}$. If Γ^ϵ consists of only one entry (label) we may ask our self how small ϵ can be made until the cardinality changes, the obtained $1-\epsilon$ is the level of confidence.

To summarize these points, the most useful prediction is those containing exactly one label. Therefore two error rates are of particular interest, ϵ_1 being the smallest ϵ and ϵ_2 being the greatest ϵ so that $|\Gamma^\epsilon| = 1$. ϵ_2 is the p-value of the best and ϵ_1 is the p-value of the second best label y. So the prediction can be summarized as

$$(\text{confidence}) = 1 - \epsilon_1 = 1 - p_{y_{2\text{nd}}} \tag{10}$$

$$(\text{credibility}) = \epsilon_2 = p_{y_{1\text{st}}} \tag{11}$$

Confidence says something about being sure that the second best label and all worse ones are wrong. Credibility says something about to be sure that the best label is right respectively that the data point is (un)typical and not an outlier.

As further pointed out in [39, 9] there are two approaches to construct conformal predictors by means of inductive or transductive learners, here we focus on transductive learners (for details see [39, 9]). While the just sketched theoretical framework of conformal prediction is a generic statistical approach, the concrete utilization needs a so called nonconformity measure which is individual for each type of algorithm.

Definition 1 (Nonconformity measure) *A nonconformity measure is a function* $A : B \times Z \to \mathcal{R}$ *With* B *as the set of all finite bags of elements in* Z*.*

In practical applications A is chosen such that large values of $A(B, z)$ indicate that z is strange relative to B. As an example for classification suppose $\mathbb{D} = \mathcal{R}^k$ and Y finite. Then, a useful nonconformity measure is:

$$A(z_1, \ldots, z_n, z) = \frac{\min_{i:y_i=y} d(x_i, y)}{\min_{i:y_i \neq y} d(x_i, y)}$$

where d refers to an arbitrary distance measure. A 95% confidence region for y_n is constructed by a nonconformity measure A, old examples $z_1, \ldots, z_{n-1}$ and a new object x_n. The procedure can be summarized as follows. We consider each labeling $y \in Y$ with B a bag consisting of $z_1, \ldots, z_{n-1}$ together with x_n, y. Let now B^{-i} the bag obtained by removing z_i, further define $W_i = A(B^{-i}, z_i)$ with $i = 1, \ldots, n$ and set:

$$p_y = \frac{\#i = 1, \ldots, n | W_i \geq W_n}{n}$$

Then p_y is the p-value for the current labeling y. It is the fraction of the elements in B that are at least as strange relative to the others as (x_n, y). Finally we include y in the confidence region if and only if $p_y > 0.05$.

4.1.2 Conformal prediction with prototype based classifiers

GLVQ and variants are successful prototype based learning algorithms with a winner rule in accordance or similar to the Eq. 1 used in the corresponding cost function. Multiple variants of this scheme have been presented but their common property is the existence of the distances d^+ and d^- (closest winner with the same (+) labeling or closest prototype with a different label (−)) used in the cost function to optimize the prototype positions. To transform GLVQ variants into conformal predictors a nonconformity measure has to be determined which is of the form of Def. 1

For prototype based networks one natural measure of non-conformity ($C(\mathbf{v_i}, c_i)$ for a given sample $\mathbf{v_i}$ and a given (crisp) labeling c_i is the sample margin as the distance of the data point to the closest prototype with the same label (+) normalized by the distance of this item to the closest prototype with an alternative labeling (−):

$$C(\mathbf{v_i}, c_i) \quad = d^+_{\min,\lambda}(\mathbf{w_r}, \mathbf{v_i}) / d^-_{\min,\lambda}(\mathbf{w_r}, \mathbf{v_i}) = d^+_{\min,\lambda}(\mathbf{x_r}, \mathbf{y_i}) / d^-_{\min,\lambda}(\mathbf{x_r}, \mathbf{y_i}) \tag{12}$$

Here, λ is some parametrization of the underlying distance measure d and the classifier decision is considered to be safe if the obtained non-conformity score is small - by means of a small distance of the datapoint to its closest prototype with the same labeling.

4.1.3 Confidence estimates within clinical studies

Conformal predictors require the definition of a valid nonconformity measure of the used modeling approach. In the former section such as measures have been presented for GLVQ networks which is applicable for SRNG as well. The estimation of confidence and credibility based on conformal prediction can be done by either induction or transduction. While the former is very common it has the drawback, that multiple splits in the data into hold-out-subsets are necessary. The transductive method avoids additional splits but is computationally expensive if the number of samples or the number of labels becomes (very) large.

In clinical proteomics the number of samples is typically small, in general around $50 - 500$ samples per class with a number of classes below 10. Hence a transductive approach is still applicable, avoiding unnecessary splittings of the data while keeping computations reliably effective. The number n used in the modeling should, however not become to small. Otherwise the validity of the conformal prediction will be decreased, or more precise the confidence bounds getting worse.

5 Clinical Data

Serum protein profiling is a promising approach for classification of cancer versus non-cancer samples. The data used in this paper are taken from a colorectal cancer (CRC) study and patients from healthy individuals[5]. Here it should be mentioned only that for each profile a mass spectrum is obtained within an analyzed mass-to-charge-ratio of 1500 to 3500Da. Two sample spectra are depicted in Figure 1. The data have been preprocessed as explained before using the approach published in [28]. The spectra are encoded by 416 wavelet-coefficients which leads to a data reduction of $\approx 95\%$ using the rawdata and is approximately twice the range of the number of peaks as obtained by the standard peak picking approach as proposed in [16] The preprocessing step has to be included in the crossvalidation procedure to avoid overfitting. For the considered data set it could be observed that the discriminating wavelet coefficients (with respect to the ks-test) at $p \leq 0.01$ including a p-value adjustment in accordance to bonferroni, reduce further to 101 (CRC) or 40 (LC) significant coefficients in a $5-$fold double cross validation. The wavelet method was used as mentioned in the previous section with $L = 4$.

The data set consist of 100 - colorectal cancer (CRC) and 90 - lung cancer (LC) data points. For the colorectal cancer and lung cancer study, 50 samples are taken from patients suffering from colorectal or lung cancer and the remaining samples are taken from a matched healthy control group. Colorectal cancer (CRC) is among the most common malignancies and remains a leading cause of cancer-related morbidity and mortality. It is well recognized that CRC arises from a multistep sequence of genetic alterations that result in the transformation of normal mucosa to a precursor abdomen and ultimately to carcinoma. Given the natural history of CRC, early diagnosis appears to be the most appropriate tool to reduce disease-related mortality. Currently, there is no early diagnostic test with sufficient diagnostic quality, which can be used as a routine screening tool. Therefore, there is a need for new biomarkers for colorectal cancer that can improve early diagnosis, monitoring of disease progression and therapeutic response and detect disease recurrence. Furthermore, these markers may give indications for targets for novel therapeutic strategies.

6 Experiments and Results

We focus on a supervised data analysis and reduce the dimensionality of the data by use of a problem specific wavelet analysis combined with a statistical selection criterion. We avoid statistical assumptions with respect to the underlying data sets, but take only measurement specific knowledge into account.

Hence we have a 101 and a 40 dimensional space of wavelet coefficients and we use multiple algorithms and metrics to determine classification models. We focus on the presented SRNG algorithm.

[5]Details about the data source can be obtained via Bruker Daltonik GmbH, 04109, Leipzig, Deutscher Platz 5d, Germany (km@bdal.de)

We trained in a first investigation a SRNG with 1 prototype per class which has been initialized as the mean of 30 randomly selected points from the training data, labeled by a post labeling procedure. The prototype optimization was done until convergence with an upper limit of 2000 iterations and a learning rate of $\alpha = 0.01$ using the strategy as proposed in [35] and [12]. The relevance parameters λ_i of the scaled Euclidean metric are adapted in parallel. This leads to a ranking of the input dimensions according to their importance for classification. A typical relevance profile using scaled Euclidean metric is depicted in Figure 3. The most important frequencies are indicated by high spiked (absolute) values. The depicted frequencies contribute substantially to classification accuracy and, therefore, are important for distinction of the classes. In all analyses we used a 5-fold CV in accordance to the suggestions in [21] because the number of sample is not so small and they are reliable homogeneous per group.

Considering the CRC study the SRNG models obtained at least $\approx 78\%$ cross validation accuracy in a 5-fold cross-validation. The usage of relevance learning typically improved the results by 10% such that a good prediction accuracy of around 90% could be achieved. The LC data set was found to be more complicated and the best obtained predictions are close to 80%. Considering the relevance profiles, looking for high ranked features, the data show the following picture. For the CRC study both metrics scaled functional and scaled Euclidean metric show similar profiles as depicted in Figure 3, the most significant features are consistent with findings as obtained by a standard peak based analysis. For the LC data set the situation is different. For the profile with scaled Euclidean metric most features are ranked as equally important with some minor exceptions. The most significant feature is encoding a peak not picked by the standard approaches and gives a cross validation accuracy of $\approx 78\%$ for its own using a kNN ($k = 3$) classifier on that feature. This shows that the wavelet encoding may help to reveal discriminative features and peaks not identified so far. The relevance profile on the LC data using the functional metric is a bit more diverse. The feature rankings are still similar with respect to the Euclidean profile but some features are pruned. Here different explanations are possible. For one position in the profile at around $2660Da$ a closer inspection with respect to the original data shows that this peak is the main peak of a quintet of closely located peaks. In the Euclidean relevance profile each peak got some relevance and the main peak obtained a higher relevance. In the functional metric only the right neighbor of the main peak is weighted high while the remaining neighbored peaks are pruned out. Further a correlation analysis of the intensities of the associated peak at $2670Da$ shows, that the discrimination power of this peak is similar to that of the new peak at around $2790Da$ which was pruned out in the functional metric but was most significant using the Euclidean metric. Hence the data representation of the functional metric is more sparse but similar discriminative as also visible in the crossvalidation results which are slightly better using the scaled functional norm on the LC data set. A comparison of the SRNG results using the different metrics and alternative algorithms is given in Table 1. It should be mentioned that for SVM the presented functional metric can not be applied directly because the generalized L^p distance has no inner product. A potential alternative would be the use of a Sobolev metric which mimics the functional nature of L^p distance but supports an inner product making the generation of a functional kernel possible [33].

One observes that the results are competitive with respect to other classifiers. The wavelet prepared data perform similar than a standardized peak picking approach with other parameters fixed but allow also the usage of features with complicated peak shape or smaller S/N level, which may be overseen by a standard peak picking approach. Considering the cross validation results for each data set in Table 1 it can be observed, that similar results were obtained using the different metrics. However the metrics itself show different properties. The relevance profile of the scaled Euclidean metric indicates most important

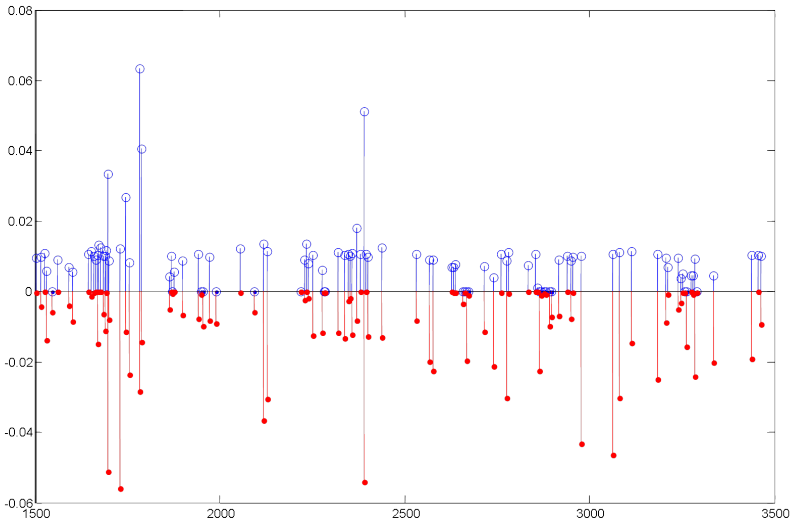


Figure 3: Visualization of a typical relevance profile obtained by SRNG using scaled Euclidean metric (upper part) and the functional norm (lower negative part of the plot) on the CRC study. Features with larger values indicate higher relevance with respect to the classification task. The x-axis indicates the relative mass position of the corresponding wavelet coefficient in the original spectrum. The y-axis is a relevance measure $\in [0, 1]$. Here relevances for the functional norm are indicated by negative values for illustration purposes.

data features in a univariate interpretation whereas the generalized L^p norm takes local neighborhoods or correlations in the data space into account while keeping the functional nature of the MS spectra. Therefore also descents in the function and not just peaks as well as correlative effects can be interpreted as relevant features. This trace of information can be further analysed by e.g. LC/MS techniques to test if a potential useful pattern can be observed which in the current linear measurement has not been sufficiently resolved so far. Beside of these good results the LVQ based approaches generates models which can be interpreted very easily by clinicians because the primal model parameters (prototypes) are representative for their receptive field. This is similar to the concept of a prototypical patient.

In Figure 5 an illustration of conformal prediction results for 20 samples of the lung cancer data set is given. The conformal prediction was done using the SRNG with the parametrized functional metric and the parameter settings as mentioned above. To interpret the shown values one should remember that high (e.g. 100%) confidence means, that all labels except the predicted one are unlikely. If say, the 10th example where predicted wrongly, this would mean that a rare event (of probability arround 1%) had occured; therefore, we expect the prediction to be correct which it is. In the case of the item 8 the confidence is also quite high (arround 90%), but we can see that the credibility is low arround 30%. From the confidence we can conclude, that the alternative label is excluded at the 10% level, but the predicted label itself is excluded at a level of around 30%. This shows, that the prediction algorithm was unable to extract from the training set enough information to allow us to confidently classify this example: the strangeness of the labeling different from the predicted label may

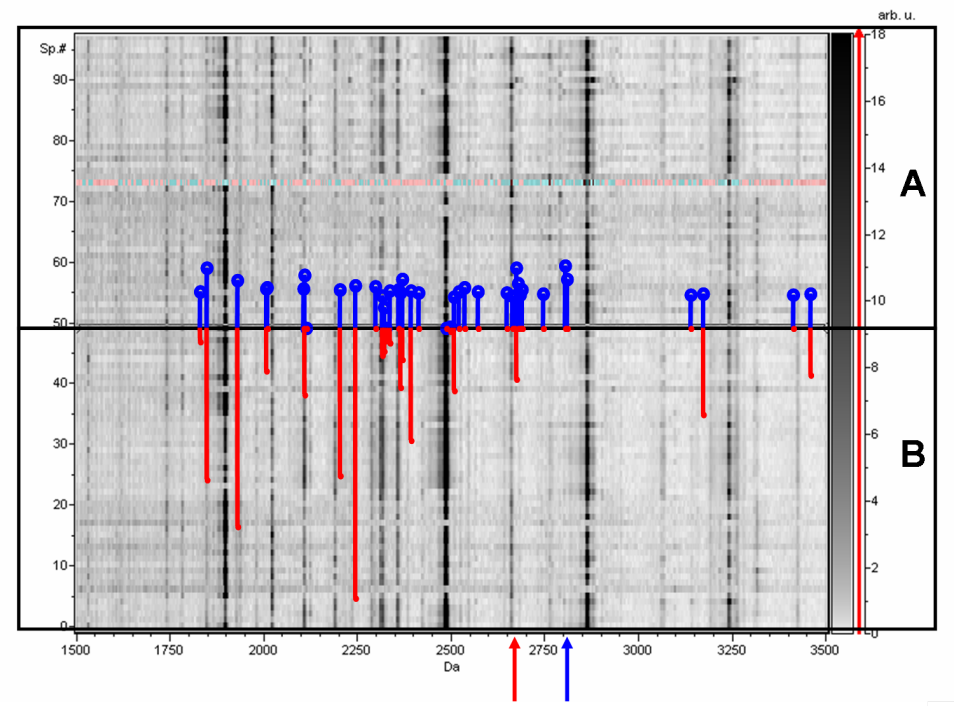


Figure 4: A gel view of the two classes (LC study) with the control class (region A) and cancer class (region B). The relevant mass positions are indicated by arrows (bottom) using the relevance profile of SRNG with scaled Euclidean metric (top overlaid plot) or functional norm (bottom overlaid plot).

be due to the fact, that the object itself is strange; perhaps the spectrum is very different from all examples in the training set. Unsurprisingly, the prediction for this example is wrong. In general, high confidence shows that all alternatives to the predicted label are unlikely. Low credibility means that the whole situation is suspect. In summary we can trust a prediction if the confidence is close to 100% and the credibility is not low (e.g. not less than 5%) [39, 9]. Taking this advice into account (with a confidence threshold of 95%) and reanalyzing the results shown in figure 5, only the items $\{4, 5, 9, 10, 15\}$ would be consideres as trusty results with high confidence and moderate or high credibility and indeed the labels for these items are correct predicted. Lowering the confidence level to 90% gives 10 trusty results, but for item 11 the prediction is wrong which means, that for this item a rare event has occurred. An analysis of further samples sets, in the way as shown in Figure 5 reveals that in general very low credibility or low confidence with high credibility, for a single item is indeed a good indicator for miss classifications, motivating the rejection of this item or assignment to the *reject* class. Which in our case of two classes should be interpreted as an unclear classification, where the considered item may belong to non of the two classes. Using the methodology of conformal prediction classification results can be judged not only on the basis of averaged cross validation accuracies but also in a fine granular single item analysis.

Initial results using the conformal prediction approach are promising. The conformal prediction on the test data sets give similar accuracy than with the standard classifiers but

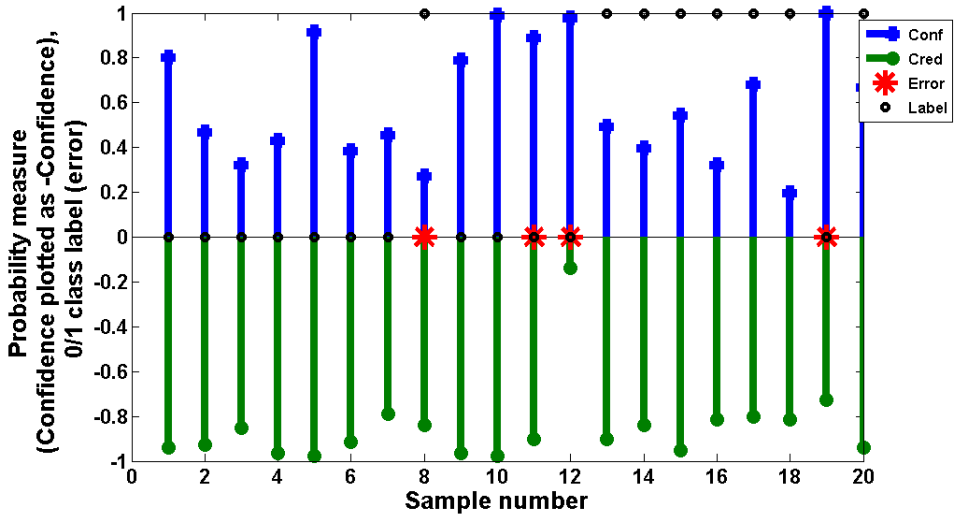


Figure 5: Visualization of conformal prediction results for 20 samples of the lung cancer data set using the parametrized functional metric. Positive entries show the values for the credibility in an obtained class prediction and negative values indicate the confidence of the single results. The predicted class labels $0, 1$ are given by black circles at 0 or 1 respectively. Miss classifications are indicated by red stars at the 0-level.

Dataset	CRC data			LC data		
Method	CV-Rec	CV-Conf	CV-Cred	CV-Rec	CV-Conf	CV-Cred
SNG-EUC	77.89%	89.28%	60.15%	75.00%	85.07%	64.15%
SNG-L^p	78.95%	89.41%	60.00%	75.00%	85.06%	64.20%
SRNG-EUC	90.53%	95.86%	53.48%	74.00%	88.52%	63.74%
SRNG-L^p	89.47%	95.68%	56.42%	78.00%	88.80%	59.67%
SVM-Linear	88.42%	*n.a.*	*n.a.*	67%	*n.a.*	*n.a.*
SVM-RBF	90.53%	*n.a.*	*n.a.*	72%	*n.a.*	*n.a.*
SVM-CPT	86.00%	*n.a.*	*n.a.*	74.00%	*n.a.*	*n.a.*
SNN-CPT	85.78%	*n.a.*	*n.a.*	72.00%	*n.a.*	*n.a.*

Table 1: Cross validated prediction accuracies, and corresponding mean confidence and credibility values for SRNG using conformal prediction and different distance measures in comparison to alternative standard approaches on wavelet encoded data. The last two rows are for comparison with the standard peak picking based approach as available in ClinProTools using default settings for SVM and SNN (a prototype classifier approach similar to SRNG).

in addition for each datapoint a confidence and credibility measure becomes available which allows a judgment of the classification decision for each single patient in a statistical manner.

7 Conclusions

We presented a specific pre-processing for mass spectrometric data analysis combined with an extension of the SRNG by a functional metric and integration of conformal prediction. The presented processing of the spectra aims on a natural compact encoding of the signals by means of a functional representation, while the classification model is especially suited to deal with high dimensional sparse data and allows strong regularizations to reduce overfitting effects.

In an initial setup the presented scenario has been embedded into a conformal prediction approach which allows the determination of clinical relevant confidence measures. The extension of conformal prediction for multiple types of prototype based classifiers has been presented.

Beside of the good results the problem of high dimensionality is still remaining. An analysis of proteomic spectra based on peak lists is in general easier to handle, e.g. it is easy to apply multiple different classification models. The wavelet based approach leads to a compact but still high dimensional representation of the data and overfitting may be a stronger issue than in contrast to a standard peak picking approach.

In future research a stronger integration of domain specific knowledge will be tried to overcome these problems and to make the approach more robust and easier to apply [6]. We will also apply the method using the priorly motivated Sobolev-Kernel[33] to improve the functional encoding using SVM.

References

[1] A. Rieder A.K. Louis, P. Maaß. *Wavelets: Theory and Applications*. Wiley, 1998.

[2] C Bishop. *Pattern Recognition and Machine Learning*. Springer Science+Business Media, LLC, New York, NY, 2006.

[3] A. Cohen, I. Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 45(5):485–560, 1992.

[4] L. P. Cordella, P. Foggia, C. Sansone, F. Tortorella, and M. Vento. Reliability parameters to improve combination strategies in multi-expert systems. *Pattern Analysis and Applications*, 2(3):205–214, 1999.

[5] C. de Stefano, C. Sansone, and M. Vento. To reject or not to reject: that is the question: an answer in case of neural classifiers. *IEEE Transactions on Systems, Man and Cybernetics Part C*, 30(1):84–93, 2000.

[6] G.M. Fiedler, S. Baumann, A. Leichtle, A. Oltmann, J. Kase, J. Thiery, and U. Ceglarek. Standardized peptidome profiling of human urine by magnetic bead separation and matrix-assisted laser desorption/ionization time-of-flight mass spectrometry. *Clinical Chemistry*, 53(3):421–428, 2007.

[6] ACKNOWLEDGMENT: The authors are grateful to T. Elssner and M. Gerhard for useful discussions and support in interpretation of the results (both Bruker Daltonik GmbH Leipzig/Bremen, Germany). Further we would like to thank Luo Zhiyuan for helpful discussions on Hedging predictions (Computer Learning Research Center (CLRC), Royal Holloway, University of London, UK). Frank-Michael Schleif would also like to thank Beate Müller (Ritsumeikan University, Japan) for an effective working atmosphere during preparation of this paper.

[7] A. Gammerman, I. Nouretdinov, B. Burford, A. Chervonenkis, V. Vovk, and Z. Luo. Clinical mass spectrometry proteomic diagnosis by conformal predictors. *Statistical applications in genetics and molecular biology*, (accepted), 2008.

[8] A. Gammerman and V. Vovk. Prediction algorithms and confidence measures based on algorithmic randomness theory. *Theoretical Computer Science*, 287:209–217, 2002.

[9] A. Gammerman and V. Vovk. Hedging predictions in machine learning. *The Computer Journal*, 50(2):151–163, 2007.

[10] N. Guerreiro, B. Gomez-Mancilla, and S. Charmont. Optimization and evaluation of seldi-tof mass spectrometry for protein profiling of cerebrospinal fluid. *Proteome science*, 4:7, 2006.

[11] B. Hammer, M. Strickert, and T. Villmann. On the generalization ability of GRLVQ networks. *Neural Proc. Letters*, 21(2):109–120, 2005.

[12] B. Hammer, M. Strickert, and T. Villmann. Supervised neural gas with general similarity measure. *Neural Proc. Letters*, 21(1):21–44, 2005.

[13] B. Hammer and Th. Villmann. Generalized relevance learning vector quantization. *Neural Netw.*, 15(8-9):1059–1068, 2002.

[14] Barbara Hammer, Marc Strickert, and Thomas Villmann. Supervised neural gas with general similarity measure. *Neural Processing Letters*, 21(1):21–44, February 2005.

[15] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, New York, 2001.

[16] R. Ketterlinus, S-Y. Hsieh, S-H. Teng, H. Lee, and W. Pusch. Fishing for biomarkers: analyzing mass spectrometry data with the new clinprotools software. *Bio techniques*, 38(6):37–40, 2005.

[17] Teuvo Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 1995. (2nd Ext. Ed. 1997).

[18] J. Lee and M. Verleysen. Generalizations of the lp norm for time series and its application to self-organizing maps. In Marie Cottrell, editor, *5th Workshop on Self-Organizing Maps*, volume 1, pages 733–740, 2005.

[19] A. Leung, F. Chau, and J. Gao. A review on applications of wavelet transform techniques in chemical analysis: 1989-1997. *Chem. and Int. Lab. Sys.*, 43(1):165–184(20), 1998.

[20] Thomas M. Martinetz, Stanislav G. Berkovich, and Klaus J. Schulten. 'Neural-gas' network for vector quantization and its application to time-series prediction. *IEEE Trans. on Neural Networks*, 4(4):558–569, 1993.

[21] A.M. Molinaro, R. Simon, and R.M. Pfeiffer. Prediction error estimation: A comparison of resampling methods. *Bioinformatics*, 21(15):3301–3307, 2005.

[22] W. Pusch, M. Flocco, S.M. Leung, H. Thiele, and M. Kostrzewa. Mass spectrometry-based clinical proteomics. *Pharmacogenomics*, 4:463–476, 2003.

[23] A. Sato and K. Yamada. Generalized learning vector quantization. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8. Proceedings of the 1995 Conference*, pages 423–9. MIT Press, Cambridge, MA, USA, 1996.

[24] E. Schäffeler, U. Zanger, M. Schwab, and M. Eichelbaum et al. Magnetic bead based human plasma profiling discriminate acute lymphatic leukaemia from non-diseased samples. In *52st ASMS Conference 2004*, page TPV 420, 2004.

[25] R. Schipper, A. loof, J. de Groot, L. Harthoorn, W. van Heerde, and E. Dransfield. Salivary protein/peptide profiling with seldi-tof-ms. *Annals of the New York Academy of Science*, 1098:498–503, 2007.

[26] F.-M. Schleif. *Prototype based Machine Learning for Clinical Proteomics*. PhD thesis, Technical University Clausthal, Technical University Clausthal, Clausthal-Zellerfeld, Germany, 2006.

[27] F.-M. Schleif, U. Clauss, Th. Villmann, and B. Hammer. Supervised relevance neural gas and unified maximum separability analysis for classification of mass spectrometric data. In *Proceedings of ICMLA 2004*, pages 374–379. IEEE Press, December 2004.

[28] F.-M. Schleif, B. Hammer, and Th. Villmann. Supervised neural gas for functional data and its application to the analysis of clinical proteom spectra. In *Proc. of IWANN 2007*, pages 1036–1044, 2007.

[29] F.-M. Schleif, M. Lindemann, P. Maass, M. Diaz, J. Decker, T. Elssner, M. Kuhn, and H. Thiele. Support vector classification of proteomic profile spectra based on feature extraction with the bi-orthogonal discrete wavelet transform. *Computing and Visualization in Science*, page in press, 2008.

[30] F.-M. Schleif, T. Villmann, and B. Hammer. Analysis of proteomic spectral data by multi resolution analysis and self-organizing-maps. In *Proc. of CIBB 2007*, pages 563–570, 2007.

[31] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. http://alrw.net (20.10.2007), 2007.

[32] V Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[33] T. Villmann. Sobolev metrics for learning of functional data - mathematical and theoretical aspects. *Machine Learning Reports*, 1(MLR-03-2007), 2007. ISSN:1865-3960 http://www.uni-leipzig.de/~compint/mlr/mlr_03_2007.pdf.

[34] T. Villmann, F.-M. Schleif, B. Hammer, and M. Kostrzewa. Exploration of mass-spectrometric data in clinical proteomics using learning vector quantization methods. *Briefings in Bioinformatics*, 9(2):129–143, 2008.

[35] T. Villmann, M. Strickert, C. Brüß, F.-M. Schleif, and U. Seiffert. Visualization of fuzzy information in in fuzzy-classification for image sagmentation using MDS. In *Proc. of ESANN 2007*, pages 103–108, 2007.

[36] Th. Villmann and B. Hammer. Supervised neural gas for learning vector quantization. In D. Polani, J. Kim, and T. Martinetz, editors, *Proc. of the 5th German Workshop on Artificial Life (GWAL-5)*, pages 9–16. Akademische Verlagsgesellschaft - infix - IOS Press, Berlin, 2002.

[37] Th. Villmann, F.-M. Schleif, and B. Hammer. Supervised neural gas and relevance learning in learning vector quantisation. In *Proceedings of WSOM 2003*, pages 47–52, 2003.

[38] Th. Villmann, F.-M. Schleif, and B. Hammer. Comparison of relevance learning vector quantization with other metric adaptive classification methods. *Neural Networks*, 19(15):610–622, 2005.

[39] V. Vovk, A. Gammerman, and G. Shafer. *Alorithmic Learning in a Random World*. Springer, New York, 2005.

[40] D.E. Waagen, M.L. Cassabaum, C. Scott, and H.A. Schmitt. Exploring alternative wavelet base selection techniques with application to high resolution radar classification. In *Proc. of the 6th Int. Conf. on Inf. Fusion (ISIF'03)*, pages 1078–1085. IEEE Press, 2003.

3.2 Supervised data analysis and reliability estimation for spectral data

The article *Supervised data analysis and reliability estimation with exemplary application for spectral data*, by F.-M.**Schleif**, T. Villmann and M. Ongyerth, appeared in NeuroComputing 72 (16-18), p. 3590-3601, in 2009. The article provides an in depth analysis of prototype based learners equipped with conformal prediction techniques for the analysis of functional data. Particular, a *thresholding approach* is proposed which can be employed in the analysis of functional spectral data combining the two measures of confidence and credibility, derived from conformal predictions. This permits to provide a reject region for prototype based learning based on a strong formalism. The approach is applied to classify remote satellite imaging data and found some novel insights in the data. I developed the thresholding algorithm and necessary experimental workflow. Experiments where done by myself and M. Ongyerth. The project was supervised by T. Villmann, who also provided the preprocessed spectral data and additional knowledge about the analysis of remote sensing data. The article was widely written by myself with specific contributions by the co-authors. All authors discussed the general article.

Additional publications in international conferences where I am co-author and which cover a similar or related topic include:

1. X. Zhu, F.-M. **Schleif**, B. Hammer, *Secure Semi-Supervised Vector Quantization for Dissimilarity Data*, In Proceedings of IWANN 2013, accepted, 2013 (Content: We extend the semi-supervised relational learning algorithm to non i.i.d. problems)

2. X. Zhu, F.-M. **Schleif**, B. Hammer, *Semi-Supervised Vector Quantization for proximity data*, In Proceedings of ESANN 2013, 89-94, 2013 (Content: Relational learning is extended by self training concepts coupled with conformal prediction)

3. F.-M. **Schleif**, X. Zhu and B. Hammer, *A conformal classifier for dissimilarity data*, In Proceedings of AIAI 2012, 234-243, 2012 (Content: Relational learning is extended by conformal prediction concepts)

4. K. Bunte, P. Schneider, B. Hammer, **F.-M. Schleif**, T. Villmann and Michael Biehl *Limited Rank Matrix Learning, discriminative dimension reduction and visualization*, In *Neural Networks*, 26, p. 159-173, 2012 (Content: Matrix learning with limited ranks, leading to an internal low dimensional representation of the data, is proposed. The method can be used to identify discriminating features, but also to get a discriminative low dimensional visualization of the data. In the experiments we used the same remote sensing data as above.)

Supervised data analysis and reliability estimation with exemplary application for spectral data

F.-M. Schleif[1]*, T. Villmann [2], M. Ongyert[3]
[1]University Leipzig, Medical Department, Leipzig, Germany
[2]University of Appl. Sc. Mittweida, Mittweida, Germany
[3]Steria Mummert Consulting AG, Leipzig, Germany

August 8, 2012

Abstract

The analysis and classification of data, is a common task in multiple fields of experimental research such as bioinformatics, medicine, satellite remote sensing or chemometrics leading to new challenges for an appropriate analysis. For this purpose different machine learning methods have been proposed. These methods usually do not provide information about the reliability of the classification. This however is a common requirement in e.g. medicine and biology. In this line the present contribution offers an approach to enhance classifiers with reliability estimates in the context of prototype vector quantization. This extension can also be used to optimize precision or recall of the classifier system and to determine items which are not classifiable. This can lead to significantly improved classification results. The method is exemplarily presented on satellite remote spectral data but is applicable to a wider range of data sets.
Keyword: spectral analysis, reliability estimation, classifier optimization, conformal prediction, rejection region, conformal thresholding

1 Introduction

The generation of classification models, is a common task in multiple fields of experimental research such as bioinformatics, medicine, satellite remote sensing or chemometrics [23, 25]. Reliability estimation of the obtained classification models is frequently required. In traditional statistics this information is usually provided by significance levels whereas for machine learning models such estimators are rare. Recently a learning theoretical approach for this problem was proposed by [33], called *conformal prediction*. We adapt this model for utilization of prototype-based classifiers like Learning Vector Quantization (LVQ) namely Supervised Relevance Neural Gas (SRNG) [32]. This model classifies each sample prototype-based and additionally offers a level of its classification reliability.

We demonstrate the capabilities of this method for classification of satellite remote sensing spectral data. For this type of data true color images allow a visual control of classification accuracy [8]. In this specific application another aspect is given by the functional character of the data which requires an adequate handling [19, 23, 29]. In particular we favor the usage of functional distances for similarity determination instead of standard euclidean metric.

*corresponding author, University Leipzig, Medical Department, email: schleif@informatik.uni-leipzig.de

The paper is organized as follows. First we briefly introduce the main ingredients for our model. We start with a short review of the Supervised Relevance Neural GAS (SRNG) for prototype based classification [32] and demonstrate how this approach can deal with different types of metrics including a functional metric. Thereafter the method of conformal prediction [33] is discussed in the light of prototype based classifiers. It is shown how a thresholding approach can be employed in the analysis of functional spectral data combining the two measures of confidence and credibility as derived from conformal predictions. The experimental settings of our approach are defined. In the experimental section we apply our framework on data obtained from remote satellite imaging. The data are analyzed in detail and some new findings are made which have not been reported so far. The paper is closed by a summary and a discussion of open points and research directions.

2 Material and Methods

2.1 Supervised Neural Gas for functional Data

Supervised Neural Gas (SNG) [10] is considered as a representative for prototype based classification approaches as introduced by Kohonen [15]. Different prototype classifiers have been proposed so far [15, 21, 10] as improvements of the original approach. The SNG combines the idea of neighborhood cooperativeness during learning from the unsupervised Neural Gas algorithm (NG)introduced in [18] with the supervised Generalized learning vector quantizer (GLVQ) as given in [21]. Subsequently we give the basic notations and some remarks to the integration of alternative metrics into Supervised Neural Gas (SNG). Details on SNG including convergence proofs can be found in [10].

Let us first clarify some notations: Let $c_{\mathbf{v}} \in \mathcal{L}$ be the label of input $\mathbf{v}$, $\mathcal{L}$ a set of labels (classes) with $\#\mathcal{L} = N_{\mathcal{L}}$. Let $V \subseteq \mathbb{R}^{D_V}$ be a finite set of inputs $\mathbf{v}$. LVQ uses a fixed number of prototypes (weight vectors, codebook vectors) for each class. Let $\mathbf{W} = \{\mathbf{w}_{\mathbf{r}}\}$ be the set of all codebook vectors and $c_{\mathbf{r}}$ be the class label of $\mathbf{w}_{\mathbf{r}}$. Furthermore, let $\mathbf{W}_c = \{\mathbf{w}_{\mathbf{r}} | c_{\mathbf{r}} = c\}$ be the subset of prototypes assigned to class $c \in \mathcal{L}$ and $\mathbf{W}_c$ is the cardinality of $\mathbf{W}_c$.

In vector quantization a stimulus vector $\mathbf{v} \in V$ is mapped onto that neuron $\mathbf{s} \in A$ the pointer $\mathbf{w}_s$ of which is closest to the presented stimulus vector $\mathbf{v}$,

$$\Psi^{\lambda}_{V \to \mathcal{A}} : \mathbf{v} \mapsto \mathbf{s}(\mathbf{v}) = \mathrm{argmin}_{\mathbf{r} \in A} d^{\lambda}(\mathbf{v}, \mathbf{w}_{\mathbf{r}}) \tag{1}$$

$d^{\lambda}(\mathbf{v}, \mathbf{w})$ is an arbitrary differentiable similarity[1] measure, which may depend on a parameter vector λ. For the moment we take λ as fixed. The neuron $\mathbf{s}(\mathbf{v})$ is called winner or best matching unit. The subset of the input space

$$\Omega^{\lambda}_{\mathbf{r}} = \{\mathbf{v} \in V : \mathbf{r} = \Psi_{V \to A}(\mathbf{v})\} \tag{2}$$

which is mapped to a particular neuron $\mathbf{r}$ according to (1), forms the (masked) receptive field of that neuron forming a Voronoi tessellation. If the class information of the weight vector is used, the boundaries $\partial\Omega^{\lambda}_{\mathbf{r}}$ generate the decision boundaries for classes. A training algorithm should adapt the prototypes such that for each class $c \in \mathcal{L}$, the corresponding codebook vectors $\mathbf{W}_c$ represent the class as accurately as possible. This means that the set of points in any given class $V_c = \{\mathbf{v} \in V | c_{\mathbf{v}} = c\}$, and the union $\mathcal{U}_c = \bigcup_{\mathbf{r}|_{\mathbf{w}_{\mathbf{r}} \in \mathbf{W}_c}} \Omega_{\mathbf{r}}$ of receptive fields of the corresponding prototypes should differ as little as possible.

[1] A similarity measure is a non-negative real-valued function, which, in contrast to a distance measure does not necessarily fulfill the triangle inequality and the symmetry property.

We suppose to have m data vectors $\mathbf{v}_i$. As pointed out in [10], the neighborhood learning for a given input $\mathbf{v}_i$ with label c is applied to the subset $\mathbf{W}_c$. The respective cost function is

$$Cost_{SNG}(\gamma) = \sum_{i=1}^{m} \sum_{\mathbf{r}|\mathbf{w_r}\in\mathbf{W}_{c_i}} \frac{h_\gamma(\mathbf{r}, \mathbf{v}_i, \mathbf{W}_{c_i}) \cdot f(\mu_\lambda(\mathbf{r}, \mathbf{v}))}{C(\gamma, K_{c_i})} \tag{3}$$

with $f(x) = (1 + \exp(-x))^{-1}$, $h_\gamma(\mathbf{r}, \mathbf{v}, \mathbf{W}) = \exp\left(-\frac{k_\mathbf{r}(\mathbf{v},\mathbf{W})}{\gamma}\right)$ and $\mu_\lambda(\mathbf{r}, \mathbf{v}) = \frac{d^\lambda_\mathbf{r} - d^\lambda_{\mathbf{r}_-}}{d^\lambda_\mathbf{r} + d^\lambda_{\mathbf{r}_-}}$ whereby $d^\lambda_{\mathbf{r}_-}$ is defined as the squared distance to the best matching prototype but labeled with $c_{\mathbf{r}_-} \neq c_\mathbf{v}$, say $\mathbf{w}_{\mathbf{r}_-}$ and $d^\lambda_\mathbf{r} = d^\lambda(\mathbf{v}, \mathbf{w}_\mathbf{r})$. For a detailed formal analysis of SNG we refer to [10].

2.1.1 Incorporation of a functional metric to SNG

As pointed out before, the similarity measure $d^\lambda(\mathbf{v}, \mathbf{w})$ is only required to be differentiable with respect to λ and $\mathbf{w}$. The triangle inequality has not to be fulfilled necessarily. This leads to a great freedom in the choice of suitable measures and allows the usage of non-standard metrics in a natural way. We now review a functional metric as given in [16]. This type of metric is especially suited in case of functional data because it takes consecutive points into account which is a natural property in case of functional data. In [16] a successful application of this type of metric was shown using the well known *tecator* data provided in [2].

The corresponding derivations can be plugged into the above equations leading to SNG with a functional metric, whereby the data are functions represented by vectors and, hence, the vector dimensions are spatially correlated. A similar situation can be observed for satellite spectra as demonstrated in [26].

Common vector processing does not take the spatial order of the coordinates into account. As a consequence, the functional aspect of spectral data is lost. For proteom spectra the order of signal features (peaks) is due to the nature of the underlying biological samples and the measurement procedure. The masses of measured chemical compounds are given ascending and peaks encoding chemical structures with a higher mass follows chemical structures with lower masses. In addition multiple peaks with different masses may encode parts of the same chemical structure and hence are correlated.

Lee proposed a distance measure taking the functional structure into account by involving the previous and next values of x_i in the i-Th term of the sum, instead of x_i alone. Assuming a constant sampling period τ, the proposed norm is:

$$\mathcal{L}_p^{FCC}(\mathbf{v}) = \left(\sum_{k=1}^{D} (A_k(\mathbf{v}) + B_k(\mathbf{v}))^p\right)^{\frac{1}{p}} \tag{4}$$

with

$$A_k(\mathbf{v}) = \begin{cases} \frac{\tau}{2}|v_k| & \text{if } 0 \leq v_k v_{k-1} \\ \frac{\tau}{2}\frac{v_k^2}{|v_k|+|v_{k-1}|} & \text{if } 0 > v_k v_{k-1} \end{cases} \tag{5}$$

$$B_k(\mathbf{v}) = \begin{cases} \frac{\tau}{2}|v_k| & \text{if } 0 \leq v_k v_{k+1} \\ \frac{\tau}{2}\frac{v_k^2}{|v_k|+|v_{k+1}|} & \text{if } 0 > v_k v_{k+1} \end{cases} \tag{6}$$

are respectively of the triangles on the left and right sides of x_i. Just as for L_p, the value of p is assumed to be a positive integer. At the left and right ends of the sequence, x_0 and x_D are assumed to be equal to zero. The derivatives for the functional metric taking $p = 2$ are given in [16]. Now we

consider the scaled functional norm where each dimension v_i is scaled by a parameter $\lambda_i > 0$ $\lambda_i \in (0, 1]$ and $\sum_i \lambda_i = 1$. Then the scaled functional norm is:

$$\mathcal{L}_p^{FCC}(\lambda \mathbf{v}) = \left(\sum_{k=1}^{D} \left(A_k(\lambda \mathbf{v}) + B_k(\lambda \mathbf{v}) \right)^p \right)^{\frac{1}{p}} \tag{7}$$

with

$$A_k(\lambda \mathbf{v}) = \begin{cases} \frac{\tau}{2} \lambda_k |v_k| & \text{if } 0 \leq v_k v_{k-1} \\ \frac{\tau}{2} \frac{\lambda_k^2 v_k^2}{\lambda_k |v_k| + \lambda_{k-1} |v_{k-1}|} & \text{else} \end{cases} \tag{8}$$

$$B_k(\lambda \mathbf{v}) = \begin{cases} \frac{\tau}{2} \lambda_k |v_k| & \text{if } 0 \leq v_k v_{k+1} \\ \frac{\tau}{2} \frac{\lambda_k^2 v_k^2}{\lambda_k |v_k| + \lambda_{k+1} |v_{k+1}|} & \text{else} \end{cases} \tag{9}$$

The corresponding derivations can be found in [26]. Using this parametrization one can emphasize/neglect different parts of the function for classification. This distance measure can be put into SNG as shown above and has been applied subsequently in the analysis of the spectra. SNG with a parametrized metric is subsequently referred as SRNG. The functional metric will be just referred as FUNC and will be always used with metric adaptation if not stated otherwise.

2.2 Conformal prediction - Reliability estimation

In the analysis of spectral data the determination of a classifier is a difficult task. The data are functional and in general high dimensional and only few assumptions about the specific nature (e.g. distributions) of the data can be made. Due to this reasons an analysis using classical statistics such as statistical tests for group comparisons, Linear discriminant analysis or partial least squares methods (see e.g. [12] for an overview) can not be applied, in general. Alternatively so called soft methods, with only minor assumptions about the specific properties of the data, are used. Typical representants of these type are prototype based classifiers such as the formerly mentioned Supervised Relevance Neural Gas [11] and variants or the famous Support Vector Machines (SVM) [28]. These methods have already proven to be appropriate for the analysis of spectral data [22, 24] also in case of very high dimensional complex problems. A drawback of these methods, in contrast to classical statistics, is the lack of reliability measures, which similar to the well known $p-$ or q-values can be used to judge the significance or reliability of a taken decisions. Only few attempts were made to give reliability estimates for these methods (see e.g. [5, 7]). Thereby the reliability estimate can be helpful to judge on the reliability of a decision but also in a more generic framework to improve the overall performance of the classifier. Reliability sometimes also referred as confidence, has been subject of a quite new theory called conformal prediction as introduced in [33]. These theory directly aims on the determination of confidence and as a second measure credibility of classifier decisions. The stability of the algorithm presented here follows immediately from the stability analysis of conformal prediction as provided in [33] because our approach is directly derived from it. According to this analysis the algorithm is stable in stochastic sense. Thereby the type of the classifier is not much limited but it is assumed that a so called *non-conformity* measure is available, revealing relevant knowledge of the classification decision. Subsequently we introduce the relevant parts of the conformal prediction approach and detail how it can be used in the analyzed experiments.

2.2.1 Settings

For the introduction to conformal prediction we switch to a more practical notation. Let the training data $z_i = (x_i, y_i) \in \mathbb{Z} = \mathbb{X} \times \mathbb{Y}$, $i = 1 \ldots L$ be given by the data points $x_i \in \mathbb{X} = \mathbb{R}^{D_V}$ and their labels

$y_i \in \mathbb{Y} = \{1, 2, \ldots, N_{\mathcal{L}}\}$ belonging to one of the $N_{\mathcal{L}}$ classes. Furthermore let x_{L+1} be a new observed data point with unknown label y_{L+1} and classification / prediction $\hat{y}_{L+1}$.

For conformal prediction we need the following terms: *conformal prediction algorithm*, *prediction region*, *nonconformity measure*,*r-value*, *error rate* ϵ, *confidence & credibility*, *exchangeability* and *validity* which are explained in more detail subsequently.

In our setting the *conformal prediction algorithm* computes for the given training data $(z_i)_{i=1,\ldots,L}$, the observed data point x_{L+1} and a chosen error rate ϵ the *prediction region* $\Gamma^\epsilon(z_1, \ldots, z_l, x_{L+1}) \subset \mathbb{Y}$ consisting of 0 to n possible labels. The applied method ensures us that if the z_i are exchangeable[2] then

$$P(y_{L+1} \notin \Gamma^\epsilon(z_1, \ldots, z_l, x_{L+1})) \leq \epsilon \tag{10}$$

holds asymptotically for $L \to \infty$ for each distribution of $\mathbb{Z}$. One says that the predictor is *asymptotically valid*. It is important to mention, that the probability is an unconditional one what means, that if we repeat the process of drawing samples x_{L+1} and generating Γ^ϵ n times we will find with respect to statistical fluctuations that in less than $\epsilon \times n$ cases the real label y_{L+1} is not under the predicted labels of Γ^ϵ. It does not mean, that for a certain x_{L+1} y_{L+1} is in Γ^ϵ with probability $> 1-\epsilon$. As counter example one considers an empty prediction region for which this conditional probability becomes exactly zero. Such cases may happen if the observed sample x_{L+1} is extremely rare in $\mathbb{X}$ ($\mathbb{Z}$) in such a way, that it is not typical with respect to the given training data. So it does not effect the error rate (10).

The conformal prediction algorithm is illustrated in Figure 1 and (11)-(15). The non conformity measure $A(D_i, z_i)$ is used to calculate a non conformity value α_i that estimates how badly z_i fits to the representative data D_i=$\{z_1, \ldots, z_{L+1}$
$z_i\}$. For a certain prediction $\hat{y}$ one calculates its *r-value* by adding $z_{L+1} = (x_{L+1}, \hat{y})$ (11) to the training data (12), calculating the α_i by checking each z_i against the rest (13) and retrieving $r_{\hat{y}}$ as the relative amount of samples that are as bad or worse conformal to all remaining examples (14). For a reasonable non conformity measure A α_{L+1} should be small if x_{L+1} is typical and the prediction $\hat{y}$ is right and typical for the data point x_{L+1}. This involves a high $r_{\hat{y}}$ and a membership of $\hat{y}$ in Γ^ϵ for most ϵ. If x_{L+1} is untypically or $\hat{y}$ is wrong A should detect this mismatch and generate a big α_{L+1}. In this case only a few examples of the training data have a greater nc-value such that $r_{\hat{y}}$ will be quiet small. As a consequence $\hat{y}$ will only be contained in Γ^ϵ for smaller ϵ.

$$\forall \hat{y} \in \mathbb{Y}$$

$$z_{L+1} \stackrel{\text{def}}{=} (x_{L+1}, \hat{y}) \tag{11}$$

$$\forall i \in \{1, \ldots L+1\}$$

$$D_i = \{z_1, \ldots, z_{L+1}\} \setminus \{z_i\} \tag{12}$$

$$\alpha_i = A(D_i, z_i) \tag{13}$$

$$r_{\hat{y}} = \frac{|\{\alpha_i : \alpha_i \geq \alpha_{L+1}|}{L+1} \tag{14}$$

$$\Gamma^\epsilon = \{y : r_{\hat{y}} > \epsilon\} \tag{15}$$

2.2.2 Non Conformity Measure

As explained in the previous section the non conformity measure should evaluate the fit of a test example z_i to representative data D_i. It is those part of the method that can incorporate detailed

[2] *exchangeability* is a weak condition: e.g. independently and identically distributed random variables are exchangeable [33]

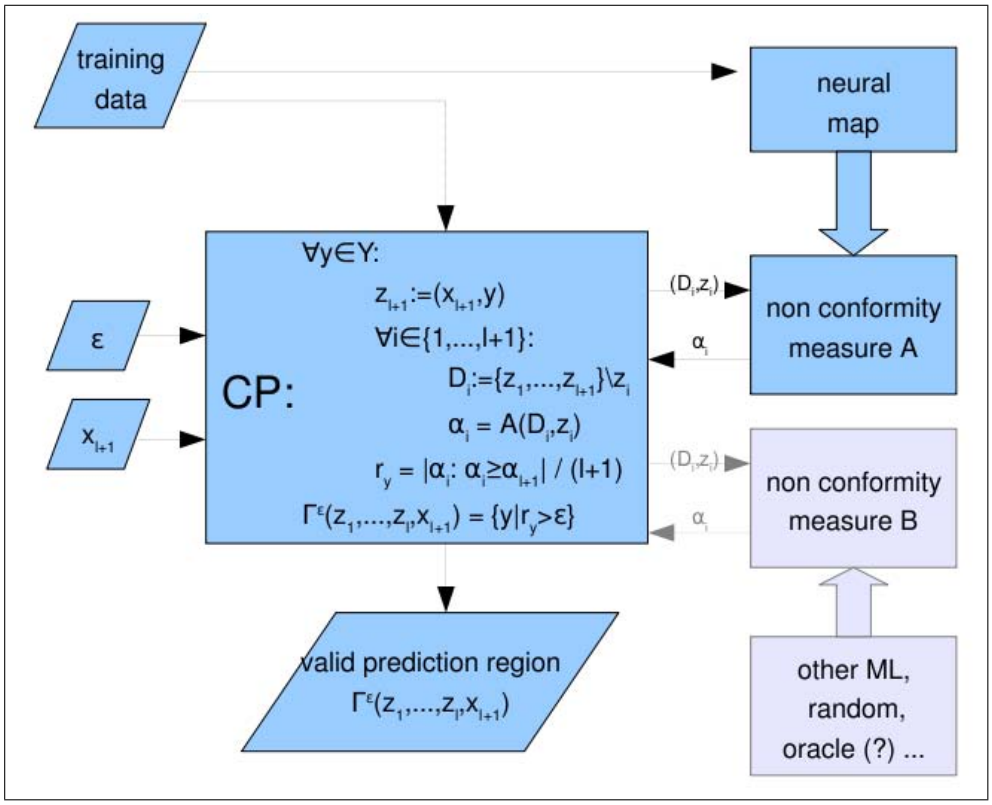


Figure 1: Schema of cooperating parts in conformal prediction. As it can be seen the conformal prediction algorithm uses the training data and an arbitrary non conformity measure to generate valid prediction regions. Normally one wants to incorporate a neural map to calculate meaningful nc values.

knowledge about the data distribution. In our setting its the place to use the learned neural map (as A in Figure 1). Nevertheless one can use any arbitrary real valued function [3] (10) but maybe with negative impact on prediction efficiency (see 2.2.3). To apply this on a prototype based situation one has to think about how the match between arbitrary z_i and D_i could be managed. A obvious solution, to learn a neural map with each individual D_i and match z_i against it, would entail high computational costs, because this has to be done for all the L *one left out* multi-sets D_i for each of the $N_{\mathcal{L}}$ test objects $(x_{L+1}, \hat{y}_{j=1,\dots,N_{\mathcal{L}}})$ in the conformal prediction algorithm. Our solution lies in the arbitrariness of A [4]. We can ignore matching z_i exactly against D_i but instead use the whole training data without z_{L+1}, therefore learning must be performed only once. The lost amount of information will be small if the number of training data is high, so that adding z_i but leaving out z_{L+1} will not change learning results dramatically.

Obvious measures for prototype based methods are k nearest neighbor methods for example:

$$\alpha_i = \frac{\sum_{j=1}^{k} d_{ij}^{+}}{\sum_{j=1}^{k} d_{ij}^{-}} \tag{16}$$

with d_{ij}^{+} being the given distance between x_i and the j-th nearest prototype with identical label y_i and d_{ij}^{-} being the given distance between x_i and the j-th nearest prototype with a label different to y_i. Other measures are conceivable.

2.2.3 Prediction Region

The prediction region $\Gamma^{\epsilon}(z_1, \dots, z_l, x_{l+1})$ stands in the center of conformal prediction. It contains for a given error rate ϵ the possible labels of $\mathbb{Y}$ that ensures (10). But how can we use it and how will it change for different values ϵ and different A?

Suppose we are using a meaningful non conformity measure A. If we would set ϵ nearly to 0 then conformal prediction has to produce Γ's that makes nearly no error at all, which can only be satisfied if Γ contains all possible labels. Of course such a prediction bears no information. But if we slowly raise ϵ we allow some rare errors to occur and as a benefit the conformal prediction algorithm excludes some unlikely labels from our prediction region and increasing its information content. In detail those $\hat{y}$ are discarded whose r-value is less equal ϵ, that means only a few z_i are as *non conformal* as $z_{L+1} = (x_{L+1}, \hat{y})$. This is a strong indicator that z_{L+1} does not belongs to the distribution $\mathbb{Z}$ and so $\hat{y}$ seams not to be the right label. If one further raises ϵ only those $\hat{y}$ will remain in Γ that can produce a high r-value meaning that the corresponding z_{L+1} is rated as very typical by A.

So one can trade the error rate against information content. The most useful prediction is those containing exactly one label. Therefore two error rates are of particular interest, ϵ_1 being the smallest ϵ and ϵ_2 being the greatest ϵ so that $|\Gamma^{\epsilon}| = 1$. ϵ_2 is the r-value of the best and ϵ_1 is the r-value of the second best label y. So the prediction can be summarized as

$$\zeta(\text{confidence}) = 1 - \epsilon_1 = 1 - r_{y_{2\text{nd}}} \tag{17}$$

$$\kappa(\text{credibility}) = \epsilon_2 = r_{y_{1\text{st}}} \tag{18}$$

Confidence says something about being sure that the second best label and all worse ones are wrong. Credibility says something about to be sure that the best label is right respectively that the data point is (un)typical and not an outlier.

As mentioned in 2.2.2 the non conformity measure has a direct impact on the efficiency of the prediction region. A good, informative measure will exclude wrong labels for small error rates and

[3] *Any* measureable function on $\mathbb{Z}^{(*)} \times \mathbb{Z}$ taking values in extended real line is called a non conformity measure

[4] This could be a constant function or a relatively to z_i fixed random value leaving out D_i at all

will reject typical data only for great error rates, meaning that $\epsilon_2 - \epsilon_1$ being large for typical data. That means, that a good measure can give useful information already for an ensured (10) small error rate ϵ_1 and on the other hand one would have to face up a high average error rate ϵ_2 to exclude the right label from the prediction region.

For practical applications here we are only interested on prediction regions with $|\Gamma^\epsilon| = 1$. For these regions natural measures of confidence and credibility become available by application of the conformal prediction methodology. These two values combined with the conformal prediction (predicted label) can be employed subsequently not only to estimate the pointwise reliability of the classification but also to improve the classifier system, by means of a thresholding approach.

2.2.4 Recall/Precision/Thresholding with Conformal Prediction

Recall-Precision graphs are very common in the field of information retrieval (IR) to estimate the performance of the considered IR-system [17]. Here we use this graphs in combination with a thresholding to improve the overall classifier performance. Thereby the recall $\mathcal{R}$ and the precision $\mathcal{P}$ are defined as:

$$\mathcal{R} = \frac{C}{L} \quad \mathcal{P} = \frac{C^+}{C} \tag{19}$$

with C as the number of classified (not rejected) data points and C^+ as the number of correct classified data points. Further we introduce a so called rejection set $\mathcal{S}_r$ and an acceptance set $\mathcal{S}_a$

$$\mathcal{S}_r = \{x_i : \zeta_i < \zeta_t \vee \kappa_i < \kappa_t\} \quad \mathcal{S}_a = \{x_i : \zeta_i \geq \zeta_t \wedge \kappa_i \geq \kappa_t\} \tag{20}$$

with ζ_t/κ_t as the user defined confidence/credibility thresholds. For a chosen threshold pair ζ_t/κ_t the definitions for recall $\mathcal{R}$ and the precision $\mathcal{P}$ are adapted in the natural way using the acceptance region such as the thresholded recall $\mathcal{R}_{(\zeta_t,\kappa_t)}$ and the thresholded precision $\mathcal{P}_{(\zeta_t,\kappa_t)}$ become:

$$\mathcal{R}_{(\zeta_t,\kappa_t)} = \frac{|\mathcal{S}_a|}{L} \quad \mathcal{P}_{(\zeta_t,\kappa_t)} = \frac{|\mathcal{S}_a|^+}{|\mathcal{S}_a|} \tag{21}$$

with $|\mathcal{S}_a|$ as the number of classified (not rejected) data points in the acceptance set and $|\mathcal{S}_a|^+$ as the number of correct classified data points in the acceptance set. An example of such a recall/precision graph for different thresholds ζ_t/κ_t is given in Figure 3.

3 Data description

We applied the algorithm to a large real world data set: a multi-spectral LANDSAT TM satellite image of the Colorado area. Airborne and satellite-borne remote sensing spectral images consist of an array of multi-dimensional vectors (spectra) assigned to particular spatial regions (pixel locations) reflecting the response of a spectral sensor at various wavelengths. A spectrum is a characteristic pattern that provides a clue to the surface material within the respective surface element. The utilization of these spectra includes areas such as mineral exploration, land use, forestry, ecosystem management, assessment of natural hazards, water resources, environmental contamination, biomass and productivity; and many other activities of economic significance [20].

Spectral images can formally be described as a matrix $\mathbf{S} = \mathbf{v}^{(x,y)}$, where $\mathbf{v}^{(x,y)} \in \mathbb{R}^{D_\mathcal{V}}$ is the vector (spectrum) at pixel location (x, y) with $D_\mathcal{V} = 6$. The description of the spectral bands is given in Table 1. The elements $v_i^{(x,y)}$, $i = 1 \ldots D_\mathcal{V}$ of spectrum $\mathbf{v}^{(x,y)}$ reflect the responses of a spectral sensor at a suite of wavelengths [4]. The spectrum is a characteristic fingerprint pattern that identifies the averaged content of the surface material within the area defined by pixel (x, y). The individual 2-dimensional image $\mathbf{S}_i = v_i{}^{(x,y)}$ at wavelength i is called the ith image band. The data density $\mathcal{P}(\mathcal{V})$

ID	frequency range	label	resolution	bits
1	0.45–0.52	blue	30×30	8
2	0.52–0.60	green	30×30	8
3	0.63–0.69	red	30×30	8
4	0.76–0.90	near IR	30×30	8
5	1.55–1.75	mid IR	30×30	8
7	2.08–2.35	mid IR	30×30	8

Table 1: Characteristics of the Landsat imaging device

may vary strongly within the data. Sections of the data space can be very densely populated while other parts may be extremely sparse, depending on the materials in the scene and on the spectral bandpasses of the sensor.

In addition to dimensionality and volume, other factors, specific to remote sensing, can make the analyses of hyperspectral images even harder. For example, given the richness of data, the goal is to separate many cover classes, however, surface materials that are significantly different for an application may be distinguished by very subtle differences in their spectral patterns. The pixels can be mixed, which means that several different materials may contribute to the spectral signature associated with one pixel. This may lead to an unsafe prediction. Training data may be scarce for some classes, and classes may be represented very unevenly (see Table 2). All the above difficulties motivate research into advanced and novel approaches. However it should be mentioned, that the presented approach is not limited to this type of application, but can be applied to a wider range of (spectral) or feature driven imaging analysis such as MALDI-Imaging [8, 30], raman spectroscopy of tissue slices or the analysis of microscopic images [3, 31] to name just a few.

The image was taken very close to colorado springs using satellites of LANDSAT-TM type[5]. The satellite produced pictures of the earth in 7 different spectral bands. The ground resolution in meter is 30×30 for the bands $1-5$ and band 7. Band 6 (thermal band) has a resolution of 60×60 only and, therefore, it is often dropped. The LANDSAT TM bands were strategically determined for optimal detection and discrimination of vegetation, water, rock formations and cultural features within the limits of broad band multi-spectral imaging. The spectral information, associated with each pixel of a LANDSAT scene is represented by a vector $\mathbf{v} \in \mathcal{V} \subseteq \mathbb{R}^{D_\mathcal{V}}$ with $D_\mathcal{V} = 6$. The aim of any classification algorithm is to subdivide this data space into subsets of data points, with each subset corresponding to specific surface covers such as forest, industrial region, etc. The feature categories are specified by prototype data vectors (training spectra). Additionally, the Colorado image is completely labeled by experts [6]. There are 14 labels describing different vegetation types and geological formations. The detailed labeling of the classes is given in Table 2, here we also specify the used coloring for the subsequently generated images as obtained from the classification models[7]. The colors where chosen such that similar materials get similar colors in the RGB space. In addition we show plots of the data using the HSV color space whereby the H channel encodes the class ($1-14$ scaled to the full range), S the results of the confidence measure ζ and V the results for the credibility measure κ. Using this setting a perfect recognition/prediction results in colors with high saturation and colorimetry (v - channel), whereas less perfect detected data points reduce the saturation and/or the colorimetry such that they appear darker and more dirty.

[5]Thanks to M. Augusteijn (University of Colorado) for providing this image.

[6]Its known that an exact ground truth labeling is complicated to obtain in this field and also effects such as the granularity may significantly effect the data and hence the label precision (e.g snow may appear as water). Under this light imprecision of the labeling is a general problem for multiple data sets.

[7]For better visualization in b/w the misclassifications are sometimes also given with white coloring. Due to some specifics of the given labeling with respect to the information encoded in the data, as pointed out in the text, the class 7 (also white) is often subject of misclassifications, anyway. Colored versions of the image can be obtained from the corresponding author.

Label	class	R	G	B	ground cover	#pixels
a	1	0	128	0	Scotch pine	581424
b	2	128	0	128	Douglas fir	355145
c	3	128	0	0	Pine / fir	181036
d	4	192	0	192	Mixed pines	272282
e	5	0	255	0	Mixed pines	144334
f	6	255	0	0	Aspen/Pines	208152
g	7	255	255	255	No veg.	170196
h	8	128	60	0	Aspen	277778
i	9	0	0	255	Water	16667
j	10	0	255	255	Moist meadow	97502
k	11	255	255	0	Bush land	127464
l	12	255	128	0	Pastureland	267495
m	13	0	128	128	Dry meadow	675048
n	14	128	128	128	Alpine veg.	27556
o	15	0	0	0	misclassif.	-

Table 2: Short description of the different classes of the satellite image, the used similarity based coloring (in RGB space) and the number of pixel present for each class.

Thereby, the label probability varies in a wide range. The size of the image is 1907×1784 pixels [8].

4 Experiments and Results

To get a valid setting of the experiments the data have been split into multiple sets, such that three data splits are obtained. These sets are named as *tuning set* (TRS) with 1500 data points per class, the *crossvalidation set* (CRS) with 3.381.079 data points has been used in a 5×5 cross validation, thereby we call each test set as the *rest set* (RS) of this crossvalidation. For the set TRS and CRS the points have been selected randomly from the original data set such that each class is equally represented. The TRS has been used for parameter tuning studies, thereby the data points have been split into a training and a test set such that 1000 points where used for training and 500 points to determine the optimal parameters. In additional experiments it has been verified that alternative set sizes of the cross-validation do not change the results significantly as long as the data statistics is sufficiently preserved. For details on this topic we refer to [1].

The parameter tuning part has been done for SRNG with standard and scaled Euclidean metric (SNG/SRNG). The identified optimal settings for the basic parameters of S(R)NG with conformal prediction were transferred to the other models. The SRNG with appropriate parametrization has been subsequently applied to the prior not used data in the CRS data set and evaluated in a 5×5-fold crossvalidation scheme. From the crossvalidation runs, showing very small variances between the different models, we choose the first model to label the whole satellite image. In the following we detail the three stages of our experiments, followed by an additional analysis employing conformal prediction in a thresholding experiment.

[8] Thereby 9 pixel have a unclear label and have been removed.

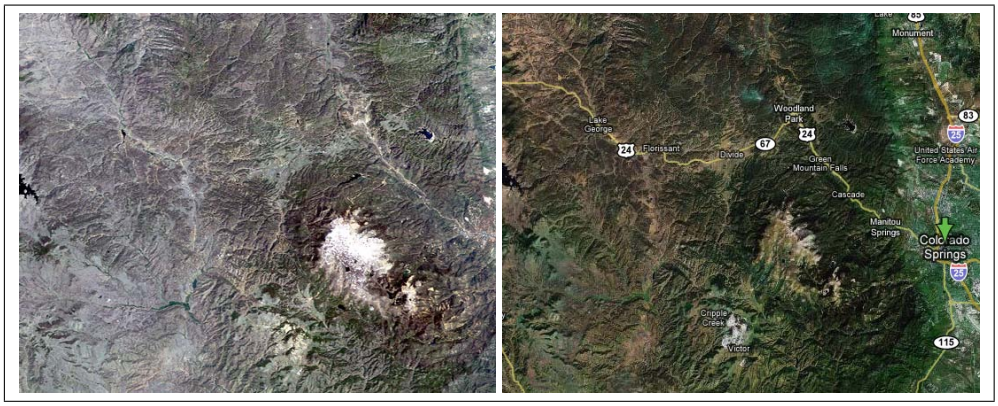


Figure 2: True coloring of the satellite data. Left the coloring in accordance to the RGB channels of the original data (data approx 1990), right a up to data image as obtained from [9]

method	$\mathbf{W}_c^5$	$\mathbf{W}_c^{10}$	$\mathbf{W}_c^{20}$	$\mathbf{W}_c^{50}$	$\mathbf{W}_c^{100}$
EUC-rec	90.50	92.1	93.7	95.0	96.5
EUC-pre	89.8	91.4	92.2	92.2	92.3
SEUC-rec	91.1	92.5	93.9	95.2	96.5
SEUC-pre	90.5	91.7	92.7	92.4	92.9

Table 3: Tuning results evaluated by recognition and prediction for metric Euc and SEUC varying the map size parameter of SRNG.

4.1 Parameter tuning

As already mentioned the SRNG parameters have been optimized on a very small subset of the original data set using the TRS split. Thereby the following parameters have been subject of optimization: map size, as the number $\mathbf{W}_c$ of prototypes per class in a range of $\{5, 10, 20, 50, 100\}$ and the parameter k of the k-NN based non-conformity measure. The remaining parameters of SRNG have been chosen in accordance to [32] with 200 training cycles for each experiment. First we analyzed the effect of different map sizes, as shown in Figure 3 using precision/recall graphs we also took the prediction accuracy of the model (on the test set of TRS) to judge the appropriate size. We observed that for a fixed $k = 1$ of the non-conformity measure a map size of 100 would give best results. However we found also that already 10 prototypes per class constitute a similar performance, therefore a map size of 10 balancing performance and model complexity was chosen as the final setting. Fixing the model size of 10 prototypes per class we varied the parameter k of the non-conformity measure in a range of $\{1, 3, 5\}$. Again we employed the recall/precision graphs and observed that for a $k = 1$ the dispersion of the overall precision was optimal. It should be mentioned that for the other metrics these parameters have been found to be stable as well as depicted in Table 3.

The use of Recall/Precision graphs motivates the use of a threshold to balance between recall and precision (see Figure 3). This of course is very problem dependent and should probably not be automated[9]. Thereby the conformal prediction approach reports the reliability parameters (ζ, κ) for each data point as ideal candidates for thresholding. Here we determined the thresholding parameters for three points (95% recall, break-even, end) point using the first model of the crossvalidation part (a model with optimized parameters) given in Table 4. The 95% recall can be considered as a natural

[9]In principle it is possible to get an automatic threshold determination e.g. by line fitting on the recall/precision graph - but this is not the focus of this paper.

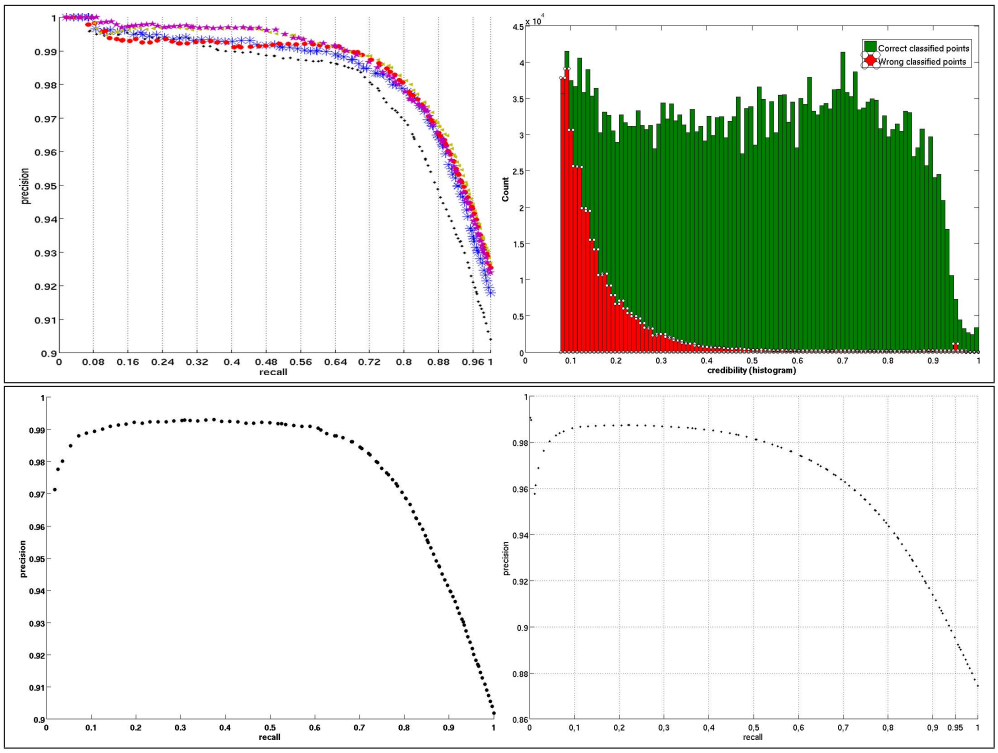


Figure 3: Recall/Precision plots for the different map sizes using SRNG (without thresholding). The curves are given as: map size 5 (dots/black), 10 (stars/blue), 20 (circle/red), 50 (filled star/magenta), 100 (arrow/yellow). The second plot shows a histogram of the credibilities determined for all data points. The third plot shows pairs of (ζ, κ) using the optimal map size 10 with $k = 1$ in the non-conformity measure for scaled Euclidean metric and the last plot a similar curve for the FUNC metric. This plot and the histogram may be employed to determined an appropriate threshold used later on.

point	(κ/ζ)
$EUC_{95\%}$	0.09/0.95
EUC_{break}	0.20/0.98
EUC_{end}	0.44/0.99
$SEUC_{95\%}$	0.12/0.92
$SEUC_{break}$	0.20/0.97
$SEUC_{end}$	0.40/0.99
$FUNC_{95\%}$	0.11/0.92
$FUNC_{break}$	0.11/0.95
$FUNC_{end}$	0.47/0.96

Table 4: Optimal thresholding parameters for (ζ, κ) as obtained by manual inspection of the recall/precision graph of one SRNG model with the different metrics.

metric	Rec.	Pred. mean	Pred. std.
EUC	n.a.	92.6	0.2
SEUC	n.a.	92.3	0.23
FUNC	n.a.	(87.4)	–

Table 5: Crossvalidation results for SRNG with metric Euc, SEUC, FUNC using the optimized parameters, without thresholding. For the FUNC metric only one model has been calculated.

criterion which allows to omit 5% of the points, occurring quite often for the analysis of real data. The second point in our analysis is the break-even point, which can be considered at that point of the recall/precision graph at which a break in the recall/precision graph can be found (e.g. a ascent of ≈ 1 for a tangent fitted against the graph). The third point is an extreme of the graph at which a further removement of points does not significantly improve the precision of the classifier[10].

The identified thresholding parameters have been used later on to get optimal precision / recall values of the classifier on the remaining (never prior used rest data RS).

4.2 Cross validation results

The SRNG with a map size of $\mathbf{W}_c = 10$ and $k = 1$ for the non-conformity measure was applied on the given satellite remote sensing data using the data subset CRS. Thereby the SRNG was trained using the three considered metrics namely, standard Euclidean metric (EUC), scaled Euclidean metric (SEUC) and the functional metric (FUNC). The results for recognition and prediction in a 5-fold crossvalidation, without thresholding, are depicted in Table 5.

One observes that the recognition and prediction accuracies are very high with close or above 90%. An analysis of the different confusion matrices supports these finding and shows also that all classes are sufficiently modeled. These findings support the results published in [32]. Interestingly the differences between the different metrics are very small. Nevertheless the metric SEUC allows the identification of discriminating features. A typical ranking of the features for SEUC is obtained as in Table 6 and visualizations of the results using the whole image are shown in 4 and 5.

[10]It should be mentioned that the generated recall/precision graph may not give a graph as a function but a cloud of distributed point. In this case we determine the convex hull of the cloud. It may also happen that the mentioned three points do not exist but only the 95% point. For our experiments it was always possible to determine the three mentioned points.

metric	D_1	D_2	D_3
SEUC	$0.08(1E^{-2})$	$0.14(2E^{-2})$	$0.24(2E^{-2})$
FUNC	0.12	0.19	0.20
metric	D_4	D_5	D_6
SEUC	$0.3(1E^{-2})$	$0.24(1E^{-2})$	0.0(0)
FUNC	0.26	0.23	0.0

Table 6: Relevance profile for the metric SEUC and FUNC. For the SEUC mean and standard deviation are shown.

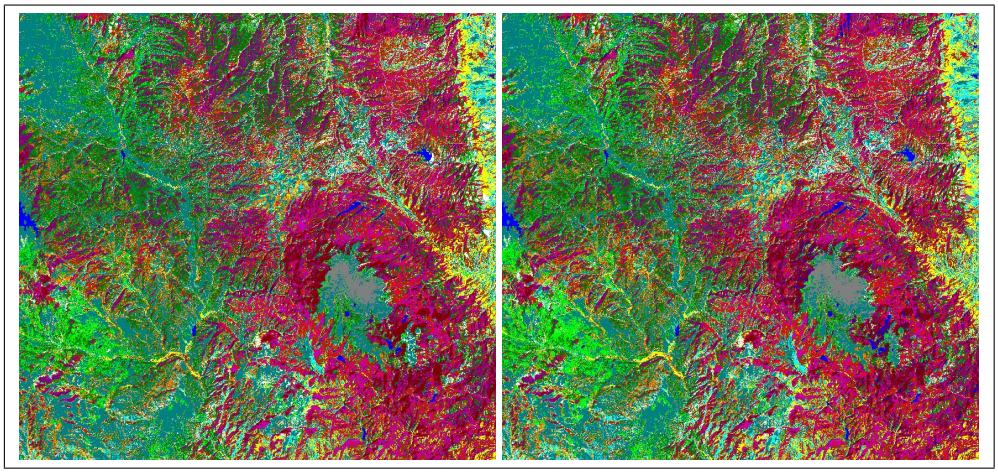


Figure 4: RGB plot for the colorado image. The left plot shows the image with the given labeling and the right plot the same image but with a predicted labeling using conformal prediction and SRNG (EUC). The color table is given as is in Table 2.

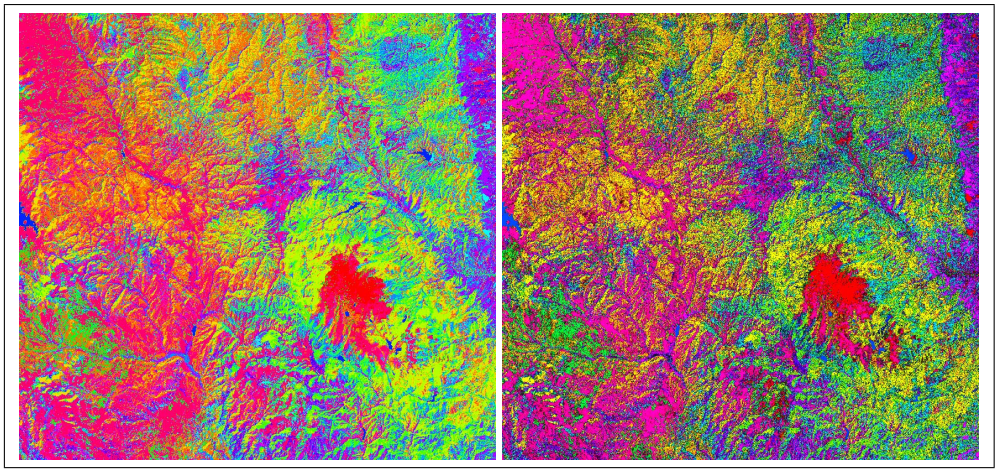


Figure 5: HSV plot for the colorado image. The left plot shows the image with the given labeling $H = labeling/14, S = 1, V = 1$ and the right plot the same image but with a predicted labeling using conformal prediction, and $S = \zeta/median(\zeta)$, $V = \kappa/median(\kappa)$ using one of the determined models. The HSV coloring is easier to interpret using conformal prediction but the coloring is not semantically related to the ground material.

point	Recall/Precision (CRS-1)	Recall/Precision (RS)
$EUC_{95\%}$	95.05/94.59	93.89/93.18
EUC_{break}	79.52/97.99	76.49/97.64
EUC_{end}	56.93/98.90	52.49/99.07
$SEUC_{95\%}$	95.18/92.20	91.54/93.08
$SEUC_{break}$	79.49/97.05	77.80/97.01
$SEUC_{end}$	60.72/99.04	60.45/98.93
$FUNC_{95\%}$	95.11/91.02	94.15/89.83
$FUNC_{break}$	77.11/95.07	79.07/94.64
$FUNC_{end}$	31.60/98.68	48.38/98.24

Table 7: Recall and precision values by application of the thresholding on SRNG-EUC, SRNG-SEUC and SRNG-FUNC using different thresholds for (ζ, κ). It can be seen that there is strong difference between the metrics but the SRNG-FUNC metric performs slightly worse than the others. As expected a more restrictive threshold (reducing the recall) improves the precision up to 99% in this case. Thereby also in case of a larger number of assignments to the *unclassified* state (EUC_{break}, $SEUC_{break}$, $FUNC_{break}$) the structural information of the satellite image is still kept as shown in Figure 6.

4.3 Thresholding

While the results found so far are already very promising we were looking for further improvements as well as a more detailed reliability estimation than plain cross validation accuracies or confusion matrices. Therefore we employed the conformal prediction methodology on the remaining test sets RS. The results for recall and precision using the different threshold are given in Table 7 for comparison the thresholding was also applied on the data used in for the first cross validation.

Multiple results can be found in the thresholding approach by considering Table 7 as well as the HSV plots on the differently thresholded RS data (see Figure 6). As a first point we see, that the thresholding improves the precision, not only on the CRS-1 data, which is expected, but also on the prior not used RS data. This observation is valid for all three thresholding points. Considering the Figure 6 we find that removing 50% of the data points still keeps the structural information encoded in the image. The removed points are in general located at the class boundaries which are a natural source of uncertainty with respect to the classification decision. The points removed at the $EUC_{95\%}$ level, again mainly account for class border points but there is also a significant amount of points which appear to be inside of classes. In fact confidence and credibility of the points are in general quite high. This however implies that the classifier was quite sure in its decision, nevertheless these points have been found to be classified wrong. A closer inspection of these points reveals that the most of it belong to the class 7 which is *no vegetation* but are classified to class 14 *alpine vegetation* (not vice versa), this is surprising but considering Figure 2 (left) the effect becomes clear. Miss classification to class 14 do always occur where the true-color image shows snow-coverage, this is due to the fact that the region labeled as *alpine vegetation* (class 14) is completely covered by snow at the time point of taking the satellite image. Hence class 14 should - with respect to the measured data - better be labeled as *snow* than *alpine vegetation*. The effect is depicted in more detail in Figure 7. There it also becomes visible, that this error in the labeling accounts for a larger number of misclassifications. Considering this case high values of confidence and credibility combined with misclassifications maybe in fact an indicator for a wrong labeling or contradictory data (see also Figure 8).

A further region of interesting points is depicted in Figure 9, nearby the Lake George (see Figure 2 (right)).

Thereby multiple misclassifications of class 3 (pine/fir) and class 2 (Douglas fir) to class 9 (water)

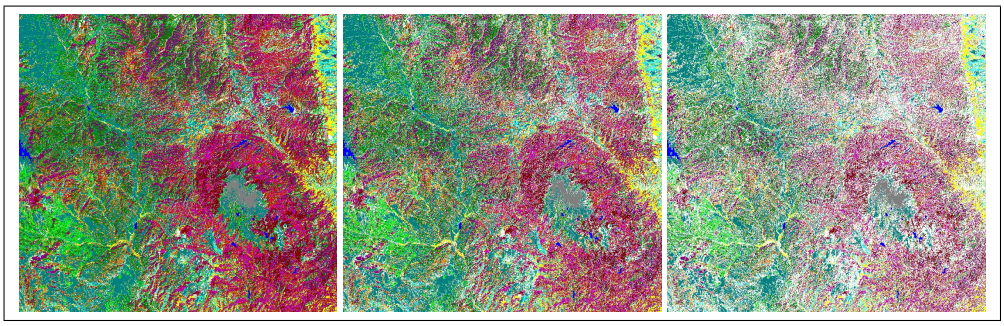


Figure 6: Visualization of the thresholding using SRNG-EUC with different threshold. Its clearly visible that the borders of the classes are subject of uncertainty but also (as pointed out later on) different interesting findings can be made with respect to the safety of a classification considering different thresholds. The first plot shows the classified (recall) pixel at a threshold of $\text{EUC}_{95\%}$, the second plot EUC_{break} and the third for EUC_{end} respectively. It can be seen that the number of rejected points (assigned to class 15 - colored white) is increasing. This helps to identify regions which are safe or unsafe with respect to the classification even if the predicted labeling is still correct.

have been found. Considering both images in Figure 2 we found that the effected pine/fir points are near to water regions. Figure 2 (right) suggests that water level may have changed and hence this miss classification are explainable also.

5 Conclusions

A method for the reliability estimation and optimization of prototype based classifiers has been presented. Thereby the approach incorporates conformal prediction to determine a threshold based on recall/precision analysis and to get reliability estimates for the classification of single items. By use of these measures the performance of the classifier can be tailored with respect to optimal recall and / or precision. This in general improves the interpretability of the generated classifications as shown here exemplarily for satellite remote sensing images. Further a classification can be analyzed with respect to its reliability and also the state of *not classifiable* can be supported. Especially the new class of unclassifiable entries is relevant in multiple classification tasks such as cases involving a classifier based automatic labeling of samples from medicine [26, 27], psychology [13, 14] or bio security domains [6], to name just a few. In these fields the confidence of the classification plays an essential role and the proposed approach offers a better interpretability of the results. In a next step the method will be applied to larger cohorts of spectral data obtained from MALDI-Imaging experiments [30]. Beside of the different promising aspects of the methods there are also some points which could be improved. Currently the choice and parametrization of the non-conformity measure must be optimized by crossvalidation a procedure which is only possible if a sufficient amount of samples is available. In future work, different non-conformity measures should be analyzed with respect to their properties under different conditions to get more generic knowledge about the behavior of a chosen measure. This knowledge could be used to simplify the formerly mentioned parametrization and choice.

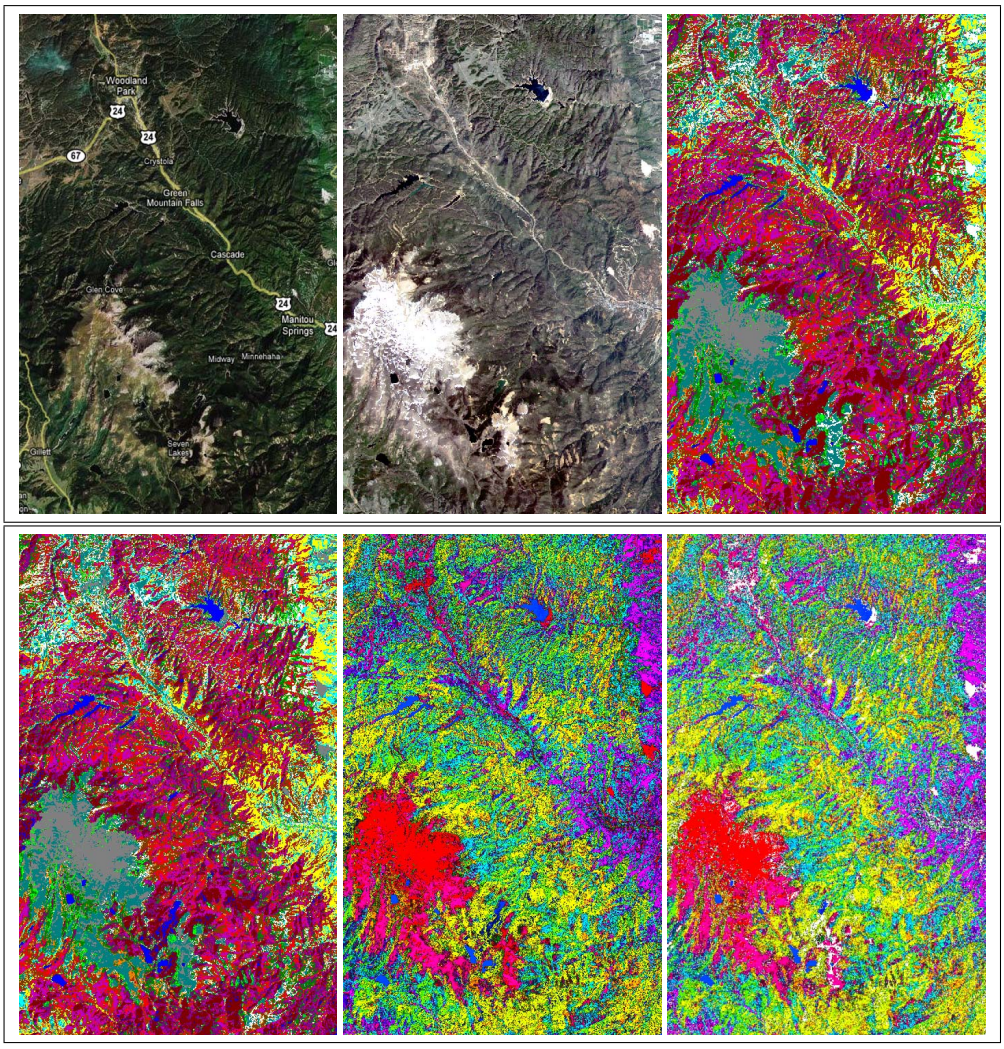


Figure 7: Region with stronger misclassifications related to the alpine-vegetation class (14). Top row shows a zoom into the region close to the alpine region. Left up to date image of the region, next true color view of this regions dating back to approximately 1990, third plot with the RGB coloring of the original labeling of the map. The second row shows the results as obtained by SRNG-EUC with conformal prediction. The plot on the left shows a coloring in RGB with the conformal predictions, the plot in the middle the HSV image using confidence and credibility. Only few dark regions (low credibility/confidence) can be found in the lower part of second plot, second row. Interestingly these items (class 12 pastureland) are not misclassified but only unsafe. But there are also regions of high confidence/credibility which are labeled as class 14 or class 7 (vegetation free).

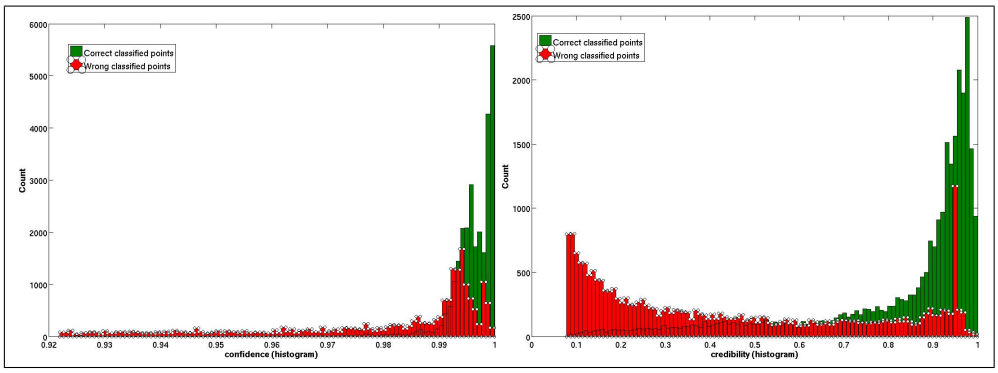


Figure 8: Confidence and credibility histogram plots. The plot helps to identify regions of high confidence with respect to the classification decision. It is also visible that there exist a larger amount with high confidence but wrong labeling - which fits to the findings presented in Figure 7.

References

[1] M. Aupetit. Homogeneous bipartition based on multidimensional ranking. In *Proc. Of European Symposium on Artificial Neural Networks (ESANN'2008)*, pages 259–264, Evere, Belgium, 2008. d-side publications.

[2] C. Blake and C. Merz. UCI repository of machine learning databases., 1998. available at: http://www.ics.uci.edu/ mlearn/MLRepository.html.

[3] C. Brüß, F. Bollenbeck, F.-M. Schleif, W. Weschke, T. Villmann, and U. Seiffert. Fuzzy image segmentation with fuzzy labelled neural gas. In *Proc. of ESANN 2006*, pages 563–569, 2006.

[4] N. W. Campbell, B. T. Thomas, and T. Troscianko. Neural networks for the segmentation of outdoor images. In *Solving Engineering Problems with Neural Networks. Proceedings of the International Conference on Engineering Applications of Neural Networks (EANN'96). Syst. Eng. Assoc, Turku, Finland*, volume 1, pages 343–6, 1996.

[5] L. P. Cordella, P. Foggia, C. Sansone, F. Tortorella, and M. Vento. Reliability parameters to improve combination strategies in multi-expert systems. *Pattern Analysis and Applications*, 2(3):205–214, 1999.

[6] National Reseach Council. *Chemical and Biological Terrorism: Research and Development to Improve Civilian Medical Response*. National Academy Press, 1999.

[7] C. de Stefano, C. Sansone, and M. Vento. To reject or not to reject: that is the question: an answer in case of neural classifiers. *IEEE Transactions on Systems, Man and Cybernetics Part C*, 30(1):84–93, 2000.

[8] S.-O. Deininger, M. Gerhard, and F.-M. Schleif. Statistical classification and visualization of maldi-imaging data. In *Proc. of CBMS 2007*, pages 403–405, 2007.

[9] Google. Free available images of colorado springs, 2008. http://maps.google.de/maps [key word: colorado springs] (last visit 17032008).

[10] B. Hammer, M. Strickert, and Th. Villmann. Supervised neural gas with general similarity measure. *Neural Processing Letters*, 21(1):21–44, 2005.

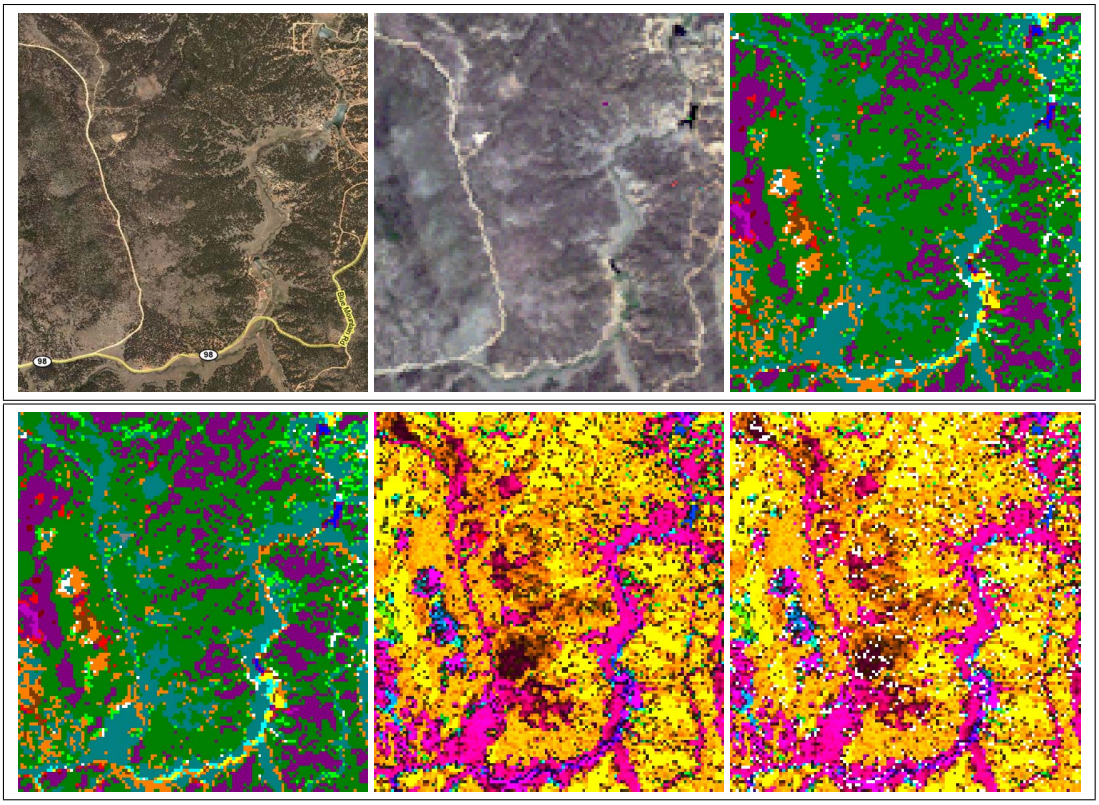


Figure 9: Region with stronger misclassifications. Top row shows a zoom into the region close to lake George. Left up to date image of the region, next true color view of this regions dating back to approximately 1990, third plot with the RGB coloring of the original labeling of the map. The second row shows the results as obtained by SRNG-EUC with conformal prediction. The plot on the left shows a coloring in RGB with the conformal predictions, the plot in the middle the HSV image using confidence and credibility. Here already dark/dirty regions can be detected indicating pixels with unsafe labeling. This is supported by an analysis of the miss classifications (right plot) where misclassified item are colored white. A closer inspection of the region using the true color maps (first row) supports these findings. The discrimination problems occur between class 13 (dry meadow - gray blue), class 12 (grass pastureland - yellow) and class 1 (Scottish fir - dark green).

[11] B. Hammer and Th. Villmann. Generalized relevance learning vector quantization. *Neural Networks*, 15(8-9):1059–1068, 2002.

[12] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, New York, 2001.

[13] W. Hermann, B. Eggers, H. Barthel, D. Clark, Th. Villmann, S. Hesse, F. Grahmann, H.-J. Kühn, O. Sabri, and A. Wagner. Correlation between automated writing movements and striatal dopaminergic innervation in patients with Wilson's disease. *Journal of Neurology*, 249(8):1082–1087, 2002.

[14] W. Hermann, A. Wagner, H.-J. Khn, F. Grahmann, and T. Villmann. Classification of fine-motoric disturbances in Wilson's disease using artificial neural networks. *Acta Neurologica Scandinavia*, 111(6):400–406, 2005.

[15] Teuvo Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 1995. (Second Extended Edition 1997).

[16] J. Lee and M. Verleysen. Generalization of the l_p norm for time series and its application to self-organizing maps. In M. Cottrell, editor, *Proc. of Workshop on Self-Organizing Maps (WSOM) 2005*, pages 733–740, Paris, Sorbonne, 2005.

[17] C. Manning and H. Schuetze. *Foundations of Statistical Natural Language Processign*. MIT Press, London, 1999.

[18] Thomas M. Martinetz, Stanislav G. Berkovich, and Klaus J. Schulten. 'Neural-gas' network for vector quantization and its application to time-series prediction. *IEEE Trans. on Neural Networks*, 4(4):558–569, 1993.

[19] J.O. Ramsay and B. W. Silverman. *Functional Data Analysis*. Springer, New York, 2006.

[20] J. A. Richards and X. Jia. *Remote Sensing Digital Image Analysis*. Springer, New York, 1999. 3rd Ed.

[21] A. Sato and K. Yamada. Generalized learning vector quantization. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8. Proceedings of the 1995 Conference*, pages 423–9. MIT Press, Cambridge, MA, USA, 1996.

[22] F.-M. Schleif, U. Clauss, Th. Villmann, and B. Hammer. Supervised relevance neural gas and unified maximum separability analysis for classification of mass spectrometric data. In *Proceedings of ICMLA 2004*, pages 374–379. IEEE Press, December 2004.

[23] F.-M. Schleif, B. Hammer, and Th. Villmann. Supervised neural gas for functional data and its application to the analysis of clinical proteom spectra. In *Proc. of IWANN 2007*, pages 1036–1044, 2007.

[24] F.-M. Schleif, M. Lindemann, P. Maass, M. Diaz, J. Decker, T. Elssner, M. Kuhn, and H. Thiele. Support vector classification of proteomic profile spectra based on feature extraction with the bi-orthogonal discrete wavelet transform. *Computing and Visualization in Science*, pages DOI: 10.1007/s00791–008–0087–z, 2008.

[25] F.-M. Schleif, T. Villmann, T. Elssner, J. Decker, and M. Kostrzewa. Machine learning and soft-computing in bioinformatics - a short journey. In *Proc. of FLINS 2006*, pages 541–548, 2006.

[26] F.-M. Schleif, T. Villmann, B. Hammer, M. v. d. Werff, A. Deelder, and R. Tollenaar. Analysis of Spectral Data in Clinical Proteomics by use of Learning Vector Quantizers. In T. G. Smolinski, M. G. Milanova, and A.-E. Hassanien, editors, *Computational Intelligence in Biomedicine and Bioinformatics*, volume 1, pages 141–167. Springer, New York, NY, USA, 2008.

[27] F.-M. Schleif, T. Villmann, M. Kostrzewa, B. Hammer, and A. Gammerman. Cancer informatics by prototype networks in mass spectrometry. *Artificial Intelligence in Medicine*, page PMID:18778925, 2008.

[28] V Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[29] T. Villmann, E. Merenyi, and U. Seiffert. Machine learning approaches and pattern recognition for spectral data. In *Proc. Of European Symposium on Artificial Neural Networks (ESANN'2008)*, pages 433–444, Evere, Belgium, 2008. d-side publications.

[30] T. Villmann, F.-M. Schleif, B. Hammer, and M. Kostrzewa. Exploration of mass-spectrometric data in clinical proteomics using learning vector quantization methods. *Briefings in Bioinformatics*, 9(2):129–143, 2008.

[31] T. Villmann, M. Strickert, C. Brüß, F.-M. Schleif, and U. Seiffert. Visualization of fuzzy information in fuzzy-classification for image segmentation using MDS. In M. Verleysen, editor, *Proc. Of European Symposium on Artificial Neural Networks (ESANN'2007)*, pages 103–108, Brussels, Belgium, 2007. d-side publications.

[32] Th. Villmann, E. Merényi, and B. Hammer. Neural maps in remote sensing image analysis. *Neural Networks*, 16(3-4):389–403, 2003.

[33] V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World*. Springer, New York, 2005.

3.3 Evolving Trees for the Retrieval of Mass Spectrometry based Bacteria Fingerprints

The article *Evolving Trees for the Retrieval of Mass Spectrometry based Bacteria Fingerprints* by S. Simmuteit, F.-M.**Schleif**, T. Villmann and B. Hammer was published by Knowledge and Information Systems 25(2), p. 327-343, in 2010. In the article the Evolving Tree (ET), a Self-Organizing-Map (SOM) based, unsupervised, learning method is extended and applied to the hierarchical analysis of taxonomic, spectral data. The article provides a formal derivation of the heuristically proposed ET of Pakkanen and Oja. It provides strategies for the parameter estimation and links evaluation methods known from SOM to the ET approach. The approach is applied to spectra with an assumed underlying hierarchical structure using two data encoding approaches. The algorithm permits a log-linear identification time of the spectra fingerprints employing the learned ET structure. I suggested the basic idea of the paper and derived, together with T. Villmann the formal derivation of the ET approach. My co-author S. Simmuteit did the experimental evaluations and a prototypical implementation of the ET method. We both extended the method to the new concepts. I implemented the preprocessing methods for the peak-based spectra processing and the sparse coding. I wrote the majority of the article and integrated the approach into the original mass spec identification tool. B. Hammer and T. Villmann supervised the project. All authors discussed the general article.
Additional publications in international conferences where I am co-author and which cover a similar or related topic include:

1. S. Simmuteit, F.-M. **Schleif**, T. Villmann and M. Kostrzewa, *Hierarchical PCA using Tree-SOM for the Identification of Bacteria*, In Proceedings of the 7th International Workshop on Self Organizing Maps WSOM 2009, 272–280, ISBN:978-3-642-02396-5,2009 (Content: Provides PCA based learning for Evolving Trees and allows the identification of relevant masses for the fingerprint spectra.)

2. S. Simmuteit, F.-M. **Schleif**, T. Villmann and T. Elssner, *Tanimoto metric in Tree-SOM for improved representation of mass spectrometry data with an underlying taxonomic structure*, In Proceedings of the International Conference on Machine Learning and Applications ICMLA 2009 , 563–567, IEEE Press, ISBN:978-0-7695-3926-3, 2009 (Content: Improves the identification performance of ET by use of a taxonomic distance measure, applied to the identification of animal furs)

3. S. Simmuteit, F.-M. **Schleif** and T. Villmann, *Hierarchical evolving trees together with global and local learning for large data sets in MALDI imaging*, In Proceedings of the Workshop on Computational Systems Biology WCSB 2010, 103–106, 2010 (Content: Extends the ET concept by an additional hierarchical layer for the processing of larger data)

4. F.-M.**Schleif**, S. Simmuteit and T. Villmann, *Hierarchical deconvolution of linear mixtures of high-dimensional mass spectra in micro-biology*, In Proceedings of the International Conference on Artificial Intelligence and Applications AIA 2011, 2011 (Content: Provides a deconvolution technique for ET to decompose spectra mixtures)

Evolving Trees for the Retrieval of Mass Spectrometry based Bacteria Fingerprints

Stephan Simmuteit[1] and Frank-Michael Schleif[1]
and Thomas Villmann[2] and Barbara Hammer[3]
[1]University Leipzig, Medical Dept., 04103 Leipzig, Germany
[2]University of Appl. Sc. Mittweida, 09648 Mittweida, Germany
[3]Clausthal Uni., Dept. of CS, 38678 Clausthal, Germany

August 8, 2012

Abstract

In this paper we investigate the application of *Evolving Trees* for the analysis of mass spectrometric data of bacteria. Evolving Trees are extensions of *Self-Organizing Maps* developed for hierarchical classification systems. Therefore they are well suited for taxonomic problems like the identification of bacteria. Here we focus on three topics, an appropriate pre-processing and encoding of the spectra, an adequate data model by means of a hierarchical Evolving Tree and an interpretable visualization. First the high-dimensionality of the data is reduced by a compact representation. Here we employ sparse coding, specifically tailored for the processing of mass spectra. In the second step the topographic information which is expected in the fingerprints is used for advanced tree evaluation and analysis. We adapted the original topographic product for Self-Organizing-Maps for Evolving Trees to achieve a judgment of topography. Additionally we transferred the concept of U-matrix for evaluation of the separability of Self-Organizing-Maps to their analog in Evolving Trees. We demonstrate these extensions for two mass spectrometric data sets of bacteria fingerprints and show their classification and evaluation capabilities in comparison to state of the art techniques.

1 Introduction

The identification of bacteria in medical and biological environments by means of classical methods like gram staining is time consuming and frequently leads to mistakes in separation of species or even genus. The utilization of mass spectrometry (MS) provides a fast and reproducible way to receive bio-chemical information to identify bacteria. This approach relies on an effective experimental design and measurement as pointed out in [41].

One key algorithmic issue is the management and storage of the high-dimensional mass spectra whereas the other main topic is the classification of the fingerprints. This requires an accurate pre-processing and a reasonable classification structure to achieve adequate storage and retrieval performance. This paper concentrates on the encoding part but considers also an appropriate representation model.

Existing approaches for the identification of bacteria by means of mass spectrometry techniques (see e.g. [16]) are based on the direct comparison of spectra with manually selected reference spectra by means of peak matching including intensities as well as mass positions.

Alternative common approaches analyze the whole sequence of peptides or genes of bacteria with algorithms like Blast to get identifications with respect to known sequences stored in databases [20, 17]. These approaches need a very accurate biochemical sample preparation as well as a sufficient technical equipment, to get a large number of usable base pairs (bp), e.g. multiple 100 bps matched against a sequencing database. In parts these approaches need equipment of high resolution as shown in [46] which make the identification not only complicated and time consuming but also quite expensive. Some other approaches, which employ also machine learning algorithms like artificial neural networks, are very specific to a given problem and hence not generic in there application field, see e.g. [37].

In this paper we use data taken from a rather new measurement approach which is robust with respect to the experimental design (growing conditions of the bacteria) and can be used also with well established mass spectrometers [1]. This avoids in parts design problems as mentioned in [45, 19].

Beside of the wet-lab and measurement aspects of bacteria identification also the data analysis part is challenging. The application of MS for bacteria identification is quite new and a representation of the taxonomic (tree-) nature of bacteria is difficult. The problem of discriminating bacteria species with MS and specifically tailored data analysis approaches is described in [1]. Forero et al. use extracted features from images of bacteria to identify them [11]. Discrimination of bacteria can be done also by bio-markers based on MS spectra [29]. Most of these approaches are also based on the evaluation of the peak intensities. In case of bacteria even the peak intensities alone are an unsafe criterion. Further, the encoded peaks (line spectra) to be compared are high-dimensional vectors representing a functional relation (mass/charge to intensity). Fast and reliable investigation of line spectra, for short (LS), requires, on the one hand side, an adequate processing, which preserves the relevant information as good as possible. On the other hand, optimum data structures for classification, like trees or other shapes [9], support fast retrieval. Available approaches have insufficient identification accuracy and retrieval performance [1, 16]. One of the most critical points is also the identification time which, for traditional approaches is quite long, due to the measurement technique and the identification model, taking up to 48 hours [16].

This contribution provides new aspects for efficient information-preserving processing of line spectra by sparse coding and subsequent data-driven classification tree generation. Both parts are based on neural methods which preserve structural information in the data space. On the encoding part this is realized by using *Sparse Coding Neural Gas (SCNG)* [24, 25] and for the classification model by means of the *Evolving Tree (ET)* [32], which are both neural vector quantizers.

2 Methods

In this section we briefly provide the main algorithmic components for our MS based identification approach of bacteria. These are the encoding of the line spectra as well as the generation of a classification tree.

2.1 Sparse Coding Neural Gas

The line spectra to be classified are high-dimensional vectors with dimensionality in the range of usually thousands of mass positions. Thus, an information-preserving dimension reduction is required for a fast data processing and a reliable analysis. Simple principal component analysis (PCA) is commonly used [14] but may fail due to the non-linearity of the data [13]. Kernel PCA is a non-linear extension but it could be difficult to process new data adequately,

see [38] p. 151. Other reduction methods can be found in approximation theory. Most prominent examples are wavelet and Fourier descriptions [12]. These approaches have in common that a predefined system of basis functions is applied to encode the data. However, this choice has to be made in advance which can be sub-optimal or may result in misleading interpretations [35]. Thus a representation in terms of a data adapted set of basis functions is demanded. *Sparse Coding (SC)* offers a solution to this problem [31]. The resulting adapted set of basis functions, however, is not complete in mathematical sense, as it is known from wavelet- and Fourier-analysis. A computationally efficient realization of SC is the *Sparse Coding Neural Gas (SCNG)* [24], which is briefly introduced in the following:

We suppose that N data vectors $\mathbf{f}_k \in \mathbb{R}^D$ are available with $\|\mathbf{f}_k\| = 1$ [30]. A set of M, maybe over-complete and/or not necessarily orthogonal, basis function vectors $\phi_j \in \mathbb{R}^D$ should be used for representation of the data in form of linear combination:

$$\mathbf{f}_k = \sum_j^M \alpha_{j,k} \cdot \phi_j + \xi_k \tag{1}$$

with $\xi_k \in \mathbb{R}^D$ being the reconstruction error vector and $\alpha_{j,k}$ are the weighting coefficients with $\alpha_{j,k} \in [0,1]$, $\sum_j \alpha_{j,k} = 1$ and $\alpha_k = (\alpha_{1,k}, \ldots, \alpha_{M,k})$. The cost function E_k for $\mathbf{f}_k$ is defined as

$$E_k = \|\xi_k\|^2 - \lambda \cdot S_k \ , 0 \leq \lambda \leq 1 \tag{2}$$

which has to be minimized. Thereby λ is a control parameter balancing sparsity and the reconstruction error. It contains a regularization term S_k, which judges the sparseness of the representation chosen as

$$S_k = \sum_j g\left(\alpha_{j,k}\right) \tag{3}$$

whereby $g(x)$ is a nonlinear function like $\exp\left(-x^2\right)$, $\log\left(\frac{1}{1+x^2}\right)$, etc.. Another choice for the regularization term would be to take the entropy

$$S_k = H\left(\alpha_k\right) \tag{4}$$

of the vector α_k. We remark that minimum sparseness is achieved iff $\alpha_{j,k} = 1$ for one arbitrary j and zero elsewhere. Using this minimum scenario, optimization is reduced to minimization of the description errors $\|\xi_k\|^2$ or, equivalently, to the optimization of the basis vectors ϕ_j. As outlined above, PCA may fail due to its linear property. This might be overcome by local PCA in local partitions Ω_i of the data space. Thereby, minimum principal component analysis requires at least the determination of first principal component. Taking into account the inherent spatial arrangement of the subsets Ω_i an efficient computation of the local first principal components is possible using the Oja-learning rule [30].

In basic SCNG N prototypes $\mathbf{W} = \left\{\mathbf{w}_i \in \mathbb{R}^D\right\}$ approximate the first principal components $\mathbf{p}_i$ of the subsets Ω_i. A functional data vector $\mathbf{f}_k$ belongs to Ω_i iff its correlation to $\mathbf{p}_i$ defined by the inner product $O\left(\mathbf{w}_i, \mathbf{f}_k\right) = \langle \mathbf{w}_i, \mathbf{f}_k \rangle$ is maximum:

$$\Omega_i = \left\{ \mathbf{f}_k | i = \underset{j}{\operatorname{argmax}} \langle \mathbf{p}_j, \mathbf{f}_k \rangle \right\} \tag{5}$$

The approximations $\mathbf{w}_i$ can be obtained adaptively by Oja-learning starting with random vectors $\mathbf{w}_i$ for time $t = 0$ with $\|\mathbf{w}_i\| = 1$. Let P be the the probability density in Ω_i. Then, for each time step t a data vector $\mathbf{f}_k \in \Omega_i$ is selected according to P and the prototype $\mathbf{w}_i$ is updated by the Oja learning-rule

$$\triangle \mathbf{w}_i = \varepsilon_t O\left(\mathbf{w}_i, \mathbf{f}_k\right)\left(\mathbf{f}_k - O\left(\mathbf{w}_i, \mathbf{f}_k\right)\mathbf{w}_i\right) \tag{6}$$

with $\varepsilon_t > 0$, $\varepsilon_t \underset{t\to\infty}{\to} 0$, $\sum_t \varepsilon_t = \infty$ and $\sum_t \varepsilon_t^2 < \infty$ which is a converging stochastic process [23]. The final limit of the process is $\mathbf{w}_i = \mathbf{p}_i$ [30].

Yet, the subsets Ω_i are initially unknown. To calculate the Ω_i knowledge about the corresponding first principal components $\mathbf{p}_i$ according to (5) are needed. This problem is solved in analogy to the original neural gas in vector quantization [27]. For a randomly selected functional data vector $\mathbf{f}_k$ (according P) for each prototype the correlation $O(\mathbf{w}_i, \mathbf{f}_k)$ is determined and the rank r_i is computed according to

$$r_i(\mathbf{f}_k, \mathbf{W}) = N - \sum_{j=1}^{N} \theta\left(O(\mathbf{w}_i, \mathbf{f}_k) - O(\mathbf{w}_j, \mathbf{f}_k)\right) \tag{7}$$

counting the number of pointers $\mathbf{w}_j$ for which the relation $O(\mathbf{w}_i, \mathbf{f}_k) < O(\mathbf{w}_j, \mathbf{f}_k)$ is valid [27]. $\theta(x)$ is the Heaviside-function. Then all prototypes are updated according to

$$\triangle \mathbf{w}_i = \varepsilon_t h_\sigma(\mathbf{v}, \mathbf{W}, i)\, O(\mathbf{w}_i, \mathbf{f}_k)\left(\mathbf{f}_k - O(\mathbf{w}_i, \mathbf{f}_k)\, \mathbf{w}_i\right) \tag{8}$$

with

$$h_{\sigma_t}(\mathbf{f}_k, \mathbf{W}, i) = \exp\left(-\frac{r_i(\mathbf{f}_k, \mathbf{W})}{\sigma_t}\right) \tag{9}$$

is the so-called neighborhood function with neighborhood range $\sigma_t > 0$. Thus, the update strength of each prototype is correlated with its matching ability. Further, the temporary data subset $\Omega_i(t)$ for a given prototype is

$$\Omega_i(t) = \left\{\mathbf{f}_k | i = \underset{j}{\operatorname{argmax}} \langle \mathbf{w}_j, \mathbf{f}_k \rangle\right\} \tag{10}$$

For $t \to \infty$ the range is decreased as $\sigma_t \to 0$ and, hence, only the best matching prototype is updated in (8) in the limit. Then, in the equilibrium of the stochastic process (8) one has $\Omega_i(t) \to \Omega_i$ for a certain subset configuration which is related to the data space shape and the density P [42]. Further, one gets $\mathbf{w}_i = \mathbf{p}_i$ in the limit. Both results are in complete analogy to usual neural gas, because the maximum over inner products is mathematically equivalent to the minimum of the Euclidean distance between the vectors [21, 27].

2.2 Evolving Trees

2.2.1 Definition and Evolving Tree Learning

The 'natural' identification methodology in taxonomy/analysis of bacteria is tree structured. Therefore, in context of machine learning, shape-aware clustering approaches [34] or decision trees (DT) may come into mind. However, DTs do not integrate structural data information like data shape and density in an adequate manner during tree generation. For the considered data the labeling is in parts unsafe or even not perfectly available, hence an unsupervised setting is more appropriate. For a pure unsupervised analysis the sparsity of the data may provide a challenging task and hence it can be expect to be valuable to included additional knowledge about the data. In this line we include the structural information on the topology of the data, by means of a hierarchical structure in the modeling. An alternative to standard DTs is presented by Pakkanen et al. – the *Evolving Trees (ET)* [32]. The ET-approach is an extension of the concept of *self-organizing maps* (SOMs) introduced by Kohonen [21]. SOMs project high-dimensional vectorial data onto a predefined low-dimensional regular grid usually chosen as a hypercube. This mapping is topology preserving under certain

conditions, i.e. in case of an unviolated topology similar data points in the data space are mapped onto the same or neighbored grid nodes. For this purpose, to each node a weight vector, also called prototype, is assigned. A data point is mapped onto this node, the prototype of which is closest according to a similarity measure in the data space, usually the Euclidean distance. This rule is called winner-takes-all. In this sense, all data points mapped onto the same node fall into the same *receptive field* of this node and the respective prototype is a representative of this field.

The usual rectangular lattice as output structure is not mandatory. Other choices are possible depending on task. ETs use trees as output structures and, hence, are potentially suited for mapping of vectorial data with hierarchical substructure.

Suppose we consider an ET $\mathcal{T}$ with nodes $r \in R_{\mathcal{T}}$ (set of nodes) and root r_0 which has the depth level $l_{r_0} = 0$. A node r with depth level $l_r = k$ is connected to its successors r' with level $l_{r'} = k+1$ by directed edges $\varepsilon_{r \to r'}$ with unit length. The set of all direct successors of the node r is denoted by S_r. For $S_r = \varnothing$, the node r is called a leaf. The degree of a node r is $\delta_r = \# S_r$, here assumed to be constant δ for all nodes except the leafs. A sub-tree $\mathcal{T}_r$ with node r as root is the set off all nodes $r' \in R_{\mathcal{T}_r}$ such that there exist a directed cycle-free path $p_{r \to r'} = \varepsilon_{r \to m} \circ \ldots \circ \varepsilon_{m' \to r'}$ with $m, \ldots, m' \in R_{\mathcal{T}_r}$ and $\circ$ is the concatenation operation. $L_{p_{r \to r'}}$ is the length of path $p_{r \to r'}$, i.e. the number of concatenations plus 1. The distance $d_{\mathcal{T}}(r, r')$ between nodes r, r' is defined as

$$d_{\mathcal{T}}(r, r') = L_{p_{\hat{r} \to r}} + L_{p_{\hat{r} \to r'}} \tag{11}$$

with paths $p_{\hat{r} \to r}$ and $p_{\hat{r} \to r'}$ in the sub-tree $\mathcal{T}_{\hat{r}}$ and $R_{\mathcal{T}_{\hat{r}}}$ contains both r and r' and the depth level $l_{\hat{r}}$ is maximum for all sub-trees $\mathcal{T}_{\hat{r}'}$ which contain r and r'. A connecting path between a node r and a node r' is defined as follows: let $p_{\hat{r} \to r'}$ and $p_{\hat{r} \to r}$ be direct paths such that $L_{p_{\hat{r} \to r'}} \bullet L_{p_{\hat{r} \to r}}$ is $d_{\mathcal{T}}(r, r')$. Then $p_{r \to r'}$ is the reverse path $p_{r' \to \hat{r}} \bullet p_{\hat{r} \to r}$ and the node set of P is denoted by $\mathcal{N}_{p_{r \to r'}}$. As for usual SOMs, each node r is equipped with a prototype $\mathbf{w}_r \in \mathbb{R}^D$, provided that the data to be processed are given by $\mathbf{v} \in V \subseteq \mathbb{R}^D$. Further, we assume a differentiable similarity measure $d_V : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$. The winner detection is different from usual SOM but remains the concept of winner-takes-all. For a given subtree $\mathcal{T}_r$ with root r the *local winner* is

$$s_{\mathcal{T}_r}(\mathbf{v}) = \underset{r \in S_r}{\arg\min} \left(d_V(\mathbf{v}, \mathbf{w}_r) \right) \tag{12}$$

If $s_{\mathcal{T}_r}(\mathbf{v})$ is a leaf then it is also the overall winner node $s(\mathbf{v})$. Otherwise, the procedure is repeated recursively for the sub-tree $\mathcal{T}_{s_{\mathcal{T}_r}}$. The *receptive field* Ω_r of a leaf r (or its prototype) is defined as

$$\Omega_r = \{\mathbf{v} \in V | s(\mathbf{v}) = r\} \tag{13}$$

and the receptive field of root r' of a sub-tree $\mathcal{T}_{r'}$ is defined as

$$\Omega_{r'} = \cup_{r'' \in R_{\mathcal{T}_{r'}}} \Omega_{r''} \tag{14}$$

The adaptation of the prototypes $\mathbf{w}_r$ takes only place for those prototypes which are leafs. The others remain fixed. This learning for a randomly selected data point $\mathbf{v} \in V$ is neighborhood-cooperative as in usual SOM:

$$\triangle \mathbf{w}_r = \epsilon h_{SOM}(r, s(\mathbf{v}))(\mathbf{v} - \mathbf{w}_r) \tag{15}$$

with $s(\mathbf{v})$ is the overall winner and $\epsilon > 0$ is a small learning rate. The neighborhood function $h_{SOM}(r, r')$ is defined as a function depending on the tree distance usually of Gaussian shape

$$h_{SOM}(r, r') = \exp\left(\frac{-\left(d_{\mathcal{T}}(r, r')\right)^2}{2\sigma^2} \right). \tag{16}$$

with neighborhood range σ.

Unlike for the SOM we cannot guarantee that $s(\mathbf{v})$ is the true best matching unit (bmu) because the tree model is subject to a stochastic optimization process.

The whole ET learning is a repeated sequence of adaptation phases according to the above mentioned prototype adaptation and tree growing beginning with a minimum tree of root r_0 and its δ successors as leafs. The decision which leafs become roots of sub-trees at a certain time can be specified by the user. Subsequently for each node r a counter b_r is defined. This counter is increased if the corresponding node becomes a winner and the node is branched at threshold $\theta \in \mathbb{N}, \theta > 0$.

Possible criteria might be the variance of the receptive fields of the prototypes or the number of winner hits during the competition. The prototypes of the new leafs should be initialized in a local neighborhood of the root prototype according to d_V. Hence, the ET also can be taken as a special growing variant of SOM as it is known for example from [4].

2.2.2 Evaluation and Visualization

Since ETs are extended variants of usual SOM one can try to transfer evaluation methods known from SOMs to ETs. One important criterion is the topology preservation property. A mathematically exact definition for SOMs is given in [43]. Several methods are developed to judge the degree of topology preservation, the best known ones are the topographic product TP and the topographic function TF, [3] and [43], respectively. A detailed comparison can be found in [2]. Although the topographic product may fail in cases of strong non-linear data shapes, it is a robust estimator for the true degree of topology preservation. It, the topographic product, relates the distance between lattice nodes in the SOM grid to the respective distances between their prototypes. Hence, the method can immediately adopted for ETs. In particular we define:

The topographic product relates for each neuron the sequence of input space neighbors to the sequence of output space neighbors. It is originally defined for rectangular and hexagonal lattice structures but can easily transferred to ETs as follows:

During the computation of TP for each node r the sequences $\mathbf{n}_j^{\mathcal{T}}(r)$ and $\mathbf{n}_j^{V}(r)$ have to be determined, where $\mathbf{n}_j^{\mathcal{T}}(r)$ denotes the j-th neighbor of r, with distance measured in the tree $\mathcal{T}$, and $\mathbf{n}_j^{V}(r)$ denotes the j-th neighbor of r, with distances $d_V\left(\mathbf{w}_r, \mathbf{w}_{\mathbf{n}_j^V(r)}\right)$ evaluated in the input space V. $d_{\mathcal{T}}(r, r')$ is the tree distance defined in (11). The sequences $\mathbf{n}_j^{\mathcal{T}}(r)$ and $\mathbf{n}_j^{V}(r)$ and further averaging over neighborhood orders j and nodes r finally lead to [3]:

$$TP = \frac{1}{N(N-1)} \sum_r \sum_{j=1}^{N-1} \frac{1}{2j} \log \left(\Pi_{l=1}^{j} \frac{d_V\left(\mathbf{w}_r, \mathbf{w}_{\mathbf{n}_l^{\mathcal{T}}(r)}\right)}{d_V\left(\mathbf{w}_r, \mathbf{w}_{\mathbf{n}_l^{V}(r)}\right)} \cdot \frac{d_{\mathcal{T}}\left(r, \mathbf{n}_l^{\mathcal{T}}(r)\right)}{d_{\mathcal{T}}\left(r, \mathbf{n}_l^{V}(r)\right)} \right) \tag{17}$$

with N being the overall number of nodes in $\mathcal{T}$. TP can take positive or negative values. Only if $TP \approx 0$ is valid, the tree structure approximately matches the topology of the input data, i.e. the ET is topology preserving. In this case the ET represents the data structure adequately.

A visualization method which also allows a further evaluation of the map is the concept of $\mathbf{U}$ matrix introduced in [40], which computes the averaged distances between prototypes for neighbored lattice nodes. Here we adapt this approach for ETs $\mathcal{T}$ as well. The resulting $\mathbf{U}$-tree is denoted by $\mathbf{U}_{\mathcal{T}}$ and has entries

$$U_{\mathcal{T}}(r') = \frac{1}{1 + \#S_r} \left[\sum_{r'' \in S_r} d_V\left(\mathbf{w}_{r'}, \mathbf{w}_{r''}\right) + d_V\left(\mathbf{w}_{r'}, \mathbf{w}_r\right) \right], r' \in S_r \tag{18}$$

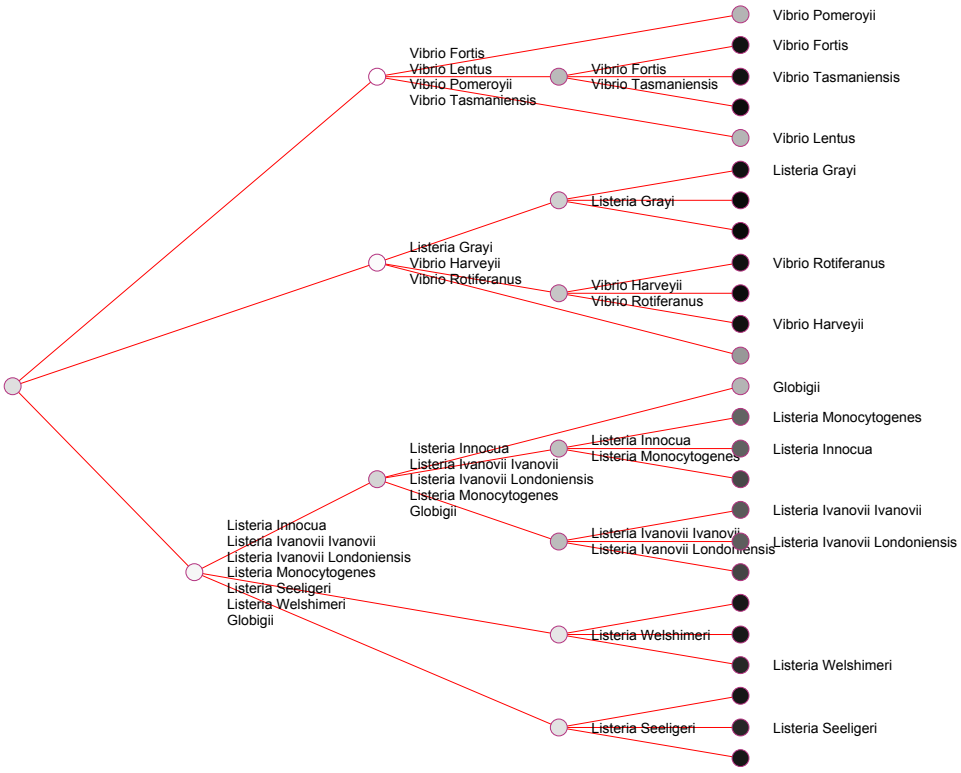


Figure 1: Labeling of nodes with the class-names of learned bacteria in the respective receptive fields. The gray-levels represent the $\mathbf{U}_{\mathcal{T}}$-tree entries, dark circles represent small values (close) and light-gray-colored circles represent high $\mathbf{U}_{\mathcal{T}}$-tree values (distant). The root node is not colored.

with S_r being again the set of direct successors of the node r. If r' is r_0, $U_{\mathcal{T}}(r') = 0$. In this way, each node is equipped with an additional value, indicating its local separation level. This value can be used in tree visualization of the ET to gray level the tree nodes accordingly. A corresponding example is depicted in Figure 1.

The $\mathbf{U}$-tree $\mathbf{U}_{\mathcal{T}}$ offers further visualization possibilities. One alternative visualization is available by a further analysis of the $\mathbf{U}_{\mathcal{T}}$ values. For this purpose we consider distances $d_{\mathbf{U}_{\mathcal{T}}(r,r')}$ with $r, r' \in \{R_{\mathcal{T}} | S_r = \emptyset, S_{r'} = \emptyset\}$, i.e. r, r' have to be leafs. Let $k^* = \max_{k \in \mathcal{N}_{p_{r \to r'}}}(\mathbf{U}_{\mathcal{T}}(k))$, the node index with maximal $\mathbf{U}$-tree-value on a path between two nodes r, r' and $\mathbf{U}_{\mathcal{T}^*} = \mathbf{U}_{\mathcal{T}}(k^*)$ the corresponding $\mathbf{U}$-tree-value then the $\mathbf{U}$-tree-value distances $d_{U_{\mathcal{T}}}(r, r')$ are given as:

$$d_{U_{\mathcal{T}}}(r, r') = {\mathbf{U}_{\mathcal{T}^*}}^{\frac{2}{l_{k^*}}} \cdot \sum_{k \in \mathcal{N}_{p_{r \to r'}}} \mathbf{U}_{\mathcal{T}}(k) \tag{19}$$

Now, with equation (19) we are able to build a matrix of $\mathbf{U}$-tree-values between all leafs and process it with an arbitrary similarity based clustering algorithm such as single-linkage clustering or relational neural gas [5] to generate a dendrogram as shown in section 3.3. The ordering of the leafs in the visualized dendrograms remains arbitrary.

A further traditional approach is to visualize the data together with such generated trees in the PCA space. The principal components of the training and the prototype data are

calculated. The obtained coefficients (PCs) can be used to draw a two- or three-dimensional illustration of training data and the connected prototypes using the first two or three most relevant PCs.

Unknown samples can be identified using the ET in the following way. The ET is fully labeled by assignment of a label to each node by an analysis of the receptive fields of the corresponding sub-trees. The root node remains unlabeled. For each receptive field a common label is determined by a majority voting of the contained samples and their labels. An unknown, new item is preprocessed as described later on. For this item the *bmu* in the tree is determined in accordance to Equation (12) and calculating $s(\mathbf{v})$. The label of the receptive field of $s(\mathbf{v})$ defines the label of the item.

3 Evolving Tree applied on Mass Spectra of Bacteria

The introduced methods are now applied to investigate MS-spectra of bacteria for classification. These data are spectra of different species of vibrio- and listeria-bacteria. Thereby we compare two kinds of pre-processed raw spectra, namely line spectra and sparse coded spectra. The ET-approach is subsequently applied to both data and the resulting classification is visualized and compared with classification according to the standard BioTyper approach.

3.1 Data

The data used in the experiments are MS spectra of 56 different vibrio species and 7 different listeria species. Every data-set contains about $20 - 40$ single spectra, being measurements of the same bacterium. Together there are 1452 spectra of vibrio and 231 spectra of listeria. Each MS measurement is processed as shown in the next section. Biological details on the bacteria samples can be obtained from [8]. For the listeria data an additional set of independent samples is available consisting of 10 measurements with an expert labeling (listeria innocua) provided by a visual evaluation of a biologist. Within the listeria data the *listeria monocytogenes* are well known to effect humans or are even pathogenic and therefore its high sensitive and specific identification is important [1].

3.2 Measurement and pre-processing

A mass spectrometer fires a laser beam onto a sample of bacteria coated with matrix solution. The material is fragmented and energized such that the fragments are accelerated in a vacuum tube into the direction of a ion-detector measuring the time-of-flight (TOF). The TOF corresponds with the mass of the sample fragment. At the end of the measurement we get a mass axis in m/z with the unit Dalton and a unit-less intensity for every mass. One obtains a high-dimensional vector (profile spectrum) of intensities, often visualized as a function of mass. More details on the mass spectrometry technique can be found in [26].

The standard way of pre-processing mass spectra to generate line spectra (consisting only of peaks) is provided by the measurement system as detailed in [7]. A line spectrum typically consists of around $100 - 500$ peaks depending on the sample complexity and system mode while the profile spectra are originally given as measurements with around 40 000 sample points. In order to map the line spectra on a common axis, the peak lists are mapped onto a global mass vector covering every appearing peak within a predefined tolerance (here 500 ppm) depending on the expected measurement accuracy.

The resulting aligned peak-lists are now located in the same data space, still very high-dimensional. For the listeria data the line spectra have a dimensionality of $D = 1181$ (peak positions) whereas for the vibrio data the dimensionality is given as $D = 2382$. An approach to achieve a further reduction of the dimensionality is sparse coding [24] as described previously. Alternative data reduction techniques for MS has been discussed in [18]. The sparse coding calculates a new basis system for the representation of the line spectra such that the contained structural information is effectively used. For this purpose the expected complexity of the structures included in the line spectra has to be incorporated into the sparse coding parameter settings. This can be done by the generation of appropriately sized patches of the original line spectra. For the listeria data 7 patches with 150 peak positions have been generated for each spectrum[1] and for the vibrio data a setting of 11 patches with 200 peak position was used. These settings have been determined by expert knowledge. In both cases the SCNG has been used with 100 prototypes to identify the new basis system. Finally we obtain 7×100 dimensions for the sparse coded listeria data and 1100 for the vibrio data, respectively.

These steps are done for the training data, but not for the test data for independent validation. We avoid the effect, that we put knowledge about the test data into the training data. For the retrieval, the data are mapped using the common mass vector obtained from the training procedure, such that we get test vectors in the same data-space as the training data. In a second step the test data are encoded in accordance to the chosen pre processing model. Either no additional processing is applied as in case of line spectra analysis, or a sparse coding in accordance to the pre-calculated sparse coding model is done.

3.3 Experimental settings

Euclidean distance is used to find the *bmu*. A number of $\delta_r = 3$ successors has been chosen for all nodes without leafs. The learning is done in accordance to the standard SOM approach, thereby the initial learning rate α_0 is defined as $\alpha_0 = 0.2$ which is exponentially decreased during learning to a final value of $\alpha_{\text{end}} = 0.01$. The neighborhood cooperation value σ is initialized with $\sigma = 1$ and exponentially decreased to $\sigma = 0.35$ in accordance to suggestions given in [43]. The total number of learning iterations I is determined depending on the number of training samples, the desired number of clusters #C, δ_r and θ in accordance to Equation (20)

$$I = \left\lceil \frac{\theta \left\lceil \frac{4 \times \#C - 1}{\delta_r - 1} \right\rceil}{|\mathrm{M}|} \right\rceil \tag{20}$$

with M as the number of samples. Equation (20) is motivated by initial studies as shown in [39]

4 Experiments and Results

Different experiments have been performed to evaluate the performance of the presented approach and to obtain reliable identifications of the considered bacterial data. We divide the experiments into two parts. First the ET is evaluated on line spectra using the vibrio or listeria data set, subsequently the same experiments are performed using a sparse coded representation. In an external validation the small independent listeria set was evaluated

[1] Due to the transformation of the spectra in patches of equal size, a small region on the border may remain which is truncated in this setting.

Threshold	Iterations	Mean Accuracy	Standard Deviation	ØNodes
496	43	82.85%	3.42	140.0
991	84	86.09%	1.80	150.0
1487	127	83.81%	2.75	151.0
1982	168	87.06%	4.22	157.0

Table 1: Crossvalidation results for 56 vibrio species (line spectra) with increasing learning time but constant expected tree size.

Threshold	Iterations	Mean Accuracy	Standard Deviation	ØNodes
153	12	100.00%	0.00	21.0
230	19	99.56%	0.77	22.0
306	24	96.93%	5.32	20.0
383	30	99.56%	0.76	21.0

Table 2: Crossvalidation results for 7 listeria species (line spectra) with increasing learning time but constant expected tree size.

on the determined ET listeria model using both codings. All experiments of the first part, in the evaluation of the ET, have been performed with different settings of the variable methods parameters and in a 3-fold cross validation to judge the classification performance and generalization capability. The parameters have been varied such that the number of cluster by means of leafs remained stable. For the first part of the experiments a comparison with the BioTyper is not possible. The available data of listeria and vibrio samples are a subset of the BioTyper data base. The BioTyper design ensures that all spectra used for modeling the BioTyper database are perfectly identified, hence a match of such data against the BioTyper will not be a fair evaluation.

The results for the vibrio study using line spectra are shown in Table 1 with different combinations of threshold and number of iterations. The threshold for the branching is provided in the first column and the number of iterations I is determined in accordance to Equation (20) leaving all other parameters fixed as mentioned priorly, to ensure an almost constant tree size.

The same experiments have been performed using listeria species as shown in Table 2.

In a further analysis vibrio and listeria data have been analyzed using ET with a sparse coding as a pre-processing step. The results of the cross-validated sparse-coded vibrio species are shown in Table 3.

Table 4 shows the result of a crossvalidation experiment with the same data like in 2, but sparse-coded.

Obviously the Evolving Tree algorithm is able to discriminate reliably bacteria species. The mean accuracy for the vibrio data using line or sparse coded spectra is at least 82% with a small standard deviation. For the listeria data the situation is different. The performance

Threshold	Iterations	Mean Accuracy	Standard Deviation	ØNodes
496	43	84.96%	3.77	152.0
991	84	82.72%	2.32	153.0
1487	127	82.36%	2.06	152.0
1982	168	84.09%	3.91	152.0

Table 3: Crossvalidation results for 56 sparse-coded vibrio species.

Threshold	Iterations	Mean Accuracy	Standard Deviation	ØNodes
153	12	87.61%	3.92	22.0
230	19	86.75%	3.23	24.0
306	24	91.88%	1.48	24.0
383	31	86.75%	5.78	24.0

Table 4: Crossvalidation results for 7 sparse coded listeria species.

	ET	BNG	KD-Tree
SC Listeria	91.88%	71.79%	94.80%
SC Vibrio	84.96%	89.16%	87.96%
LS Listeria	100.00%	50.48%	93.51%
LS Vibrio	87.06%	76.90%	90.70%

Table 5: Crossvalidation results for the Listeria and Vibrio data set. The data is preprocessed to line spectra (LS) and sparse coded data (SC). Columns show results for Evolving Tree (ET) and Batch Neural Gas (BNG) and the supervised KD-Tree.

using the line spectra approach is quite good with a mean accuracy of at least 97% and also a small standard deviation. For the same data using sparse coding the identification performance is only around 85% with a slightly larger standard deviation. Thereby the number of iterations has only a small impact on an improvement of the results and already a short learning of the ET gives reliable results. The sparse coding does not improve the identification accuracy, but leads to significantly faster identifications, because the dimensionality of the data encoded in the ET could be strongly reduced without a significant loss of identification performance.

In Table 6 we compare the identification results of the state of the art MS bacteria spectra identification tool BioTyper as provided by [6] with the Evolving Tree using the independent listeria data set. The underlying models for the BioTyper as well as for the ET were obtained using the same large listeria data set as in the previous experiments (see Table 2).

The BioTyper identifications provide a score value which is interpreted as follows: a score value of ≥ 2 indicates a secure genus identification. A score value ≥ 2.3 indicates a highly probable species identification. Lower values then 2 raises doubts on the identification, even on the genus and additional tests are recommended. Details on the scoring as well as on the BioTyper algorithm are published in [15]. The Evolving Tree has a 100% identification rate for this data-set using line spectra, but fails in the species identification by use of a sparse coding as a pre-processing step. Considering the rather unsafe results provided by the BioTyper this behavior of the ET using sparse coded data can be explained. The identification of the bacteria on a species level is obviously complicated and small difference become relevant which are lost due to the sparseness of the coding in favor of a faster identification and tree generation. Hence for bacteria which are hard to discriminate on the species level sparse coding may have limited applicability.

In case of unsafe identifications a further analysis of the data e.g. by an inspection of a corresponding tree visualization may be helpful. For the listeria data set a $\mathbf{U}_{\mathcal{T}}$-tree based visualization using the line spectra is depicted in Figure 2 and shows that the listeria species *Innocua* and *Monocytogenes* fall in the same branch and hence can be considered to be very similar. A similar situation can be observed in the Figure 1. This result is in perfect agreement with the findings shown in Table 6. In Table 6 an external validation

SpNo.	BioTyper	Score	S/H/D	ET- LS	ET - SC
1	Ivanovii Ivanovii	1.881	D	Innocua	Monocytogenes
2	Monocytogenes	1.837	D	Innocua	Monocytogenes
3	Innocua	2.029	H	Innocua	Monocytogenes
4	Innocua	1.854	D	Innocua	Monocytogenes
5	Innocua	1.948	D	Innocua	Monocytogenes
6	Innocua	1.860	D	Innocua	Innocua
7	Innocua	2.004	H	Innocua	Monocytogenes
8	Monocytogenes	1.915	D	Innocua	Monocytogenes
9	Innocua	2.205	S	Innocua	Monocytogenes
10	Innocua	2.264	S	Innocua	Monocytogenes

Table 6: Identification of unknown listeria with the Bruker BioTyper and an Evolving Tree. Column S/H/D is (S)ure, (H)igh, (D)oubt identification.

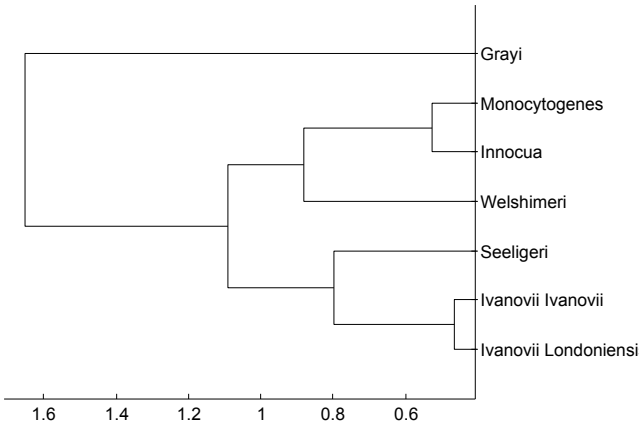


Figure 2: Evolving Tree and $\mathbf{U}_{\mathcal{T}}$-tree based dendrogram of seven listeria species

set has been used and processed as described previously. This set is not included in the BioTyper-Database and has also not been used in the model generations steps for the ET approaches. Yet, this set is completely independent and can be seen as a real live test set. All these spectra belong to the Innocua group as validated by manual analysis using traditional approaches. We observe that the BioTyper is a bit unsafe with respect to its identification. This is reflected by low score values (below 2.0) and three potential species assignments (Ivanovii Ivanovii, Monocytogenes, Innocua). For the ET approach both pre-processing techniques lead to consistent identifications. In case of line spectra the identifications were always correct and for the sparse coded data the very similar species of Monocytogenes (see 1) has been identified in general.

The time needed for a reliable identification is linear in case of the BioTyper approach and hence gets longer with an increasing database. For the ET the complexity of a query remains log-linear depending on the depth and branch-degree of the tree. For a larger number of considered types of bacteria this leads to a significantly faster identification by ET than by the BioTyper. As for a standard SOM approach the dimensionality of the data has a strong influence on the model generation as well as on the retrieval time in the distance calculations and weight updates. This complexity can effectively be reduced by application of sparse coding as an additional pre-processing step. This has only a minor impact on

the identification performance as shown above. In case of the expected huge amount of queries in real life applications the time needed for the tree generation and the encoding of new items can be neglected. To compare the performance of the presented approach with alternative techniques as shown in Table 5, experiments with the batch-Neural-Gas (BNG) [10] algorithm, which is a kind of an advanced k-means clustering, and a KD-Tree [28] implementation have been done[2]. An approach for the comparison of clustering under unsupervised settings has been presented in [33]. In our experiments the labels are provided hence a more accurate evaluation by means of cross-validation is possible. The number of prototypes for the batch-Neural-Gas have been chosen in accordance to the number of leafs in the ET, further we selected the parameter set of ET with the best performance on the training data for comparison. The KD-Tree has been applied using the Gini-Index as split-criterion. We observe for the unsupervised BNG that the ET has always a better performance on all considered data sets. For the sparse coded vibrio data the performance is comparable, but also slightly better. With respect to the encoding technique we observe that sparse coding significantly improved the performance of BNG while for ET there is no strong effect on the performance. These result clearly indicate that in an unsupervised setting the introduction of topological knowledge about the data space is helpful and provides improved identification results by ET. This is especially remarkable for the identification of bacteria, because for the taxonomy of bacteria the labeling is in parts unsafe and still subject of change - hence a stable supervised setting is in general not really available. If we consider the data in a supervised setting as for KD-Tree we observe that the identification results improved further, also with respect to ET. However the generated KD-Trees have been found to be extremely unbalanced, in general they order in a flat chain, leading to again linear query times. As already pointed out the labeling of bacteria is not always valid and hence a unsupervised approach maybe preferable as also shown in [15, 1].

5 Conclusions

In this contribution an approach for the generation and evaluation of models for the identification of bacteria spectra has been proposed. It could be shown that the ET gives reliable results in comparison to the standard BioTyper method but with significantly faster identifications due to the tree structure. The presented $\mathbf{U}_{\mathcal{T}}$-tree based visualizations give additional insights in the identification procedure and provide easy usable access to the models. It could be shown that the presented approach is high efficient with respect to the retrieval accuracy and identification speed. Coupled with the used MS based identification technique the presented method is especially interesting for bacteria identifications in the clinical domain due to time and cost restrictions and the effective method workflow needed in such an environment [22]. Thereby ET are not limited to the identification of bacteria spectra but can be applied also on other types of data as long as an appropriate pre-processing is available. Future work should include the processing of spectra, which consist of a mixture of bacteria cultures. Todays identification approaches only allow the identification of single bacteria, otherwise the identification becomes blurred and post-identification steps become mandatory. As a future extension it maybe desirable to combine the fast identification performance of the ET with the well established identification approach of the BioTyper, such that the pre identification-decision of the ET is checked against the BioTyper but only for that part of the database containing candidates of the same genus or species level. Further the ET can be extended to provide reliability estimates using the approach presented in [36]

[2] A PCA on the data with subsequent application of KD-Tree results in a performance of only 30% - this also motivates data specific encoding techniques.

such that a similar evaluation like with the BioTyper scores can be obtained. In case of taxonomic data labels for training data become available and could be used in a supervised manner during learning. This provides the opportunity to develop a supervised variant of the Evolving Tree similarly to [44][3].

References

[1] S. B. Barbuddhe, T. Maier, G. Schwarz, M. Kostrzewa, H. Hof, E. Domann, T. Chakraborty, and T. Hain. Rapid identification and typing of listeria species by matrix-assisted laser desorption ionization-time of flight mass spectrometry. *Applied and Environmental Microbiology*, 74(17):5402–5407, 2008.

[2] H.-U. Bauer, M. Herrmann, and T. Villmann. Neural maps and topographic vector quantization. *Neural Networks*, 12(4–5):659–676, 1999.

[3] H.-U. Bauer and K. R. Pawelzik. Quantifying the neighborhood preservation of Self-Organizing Feature Maps. *IEEE Trans. on Neural Networks*, 3(4):570–579, 1992.

[4] H.-U. Bauer and T. Villmann. Growing a Hypercubical Output Space in a Self–Organizing Feature Map. *IEEE Transactions on Neural Networks*, 8(2):218–226, 1997.

[5] B.Hammer and A.Hasenfuss. Relational neural gas. *Künstliche Intelligenz*, pages 190–204, 2007.

[6] Bruker Daltonik GmbH. Bruker BioTyper 2.0. Available via http://www.bdal.de, 2008.

[7] Bruker Daltonik GmbH. Bruker BioTyper 2.0, user manual. Available via http://www.bdal.de, 2008.

[8] Bruker Daltonik GmbH. Bruker listeria and vibrio spectra. Available via http://www.bdal.de (Dr. Markus Kostrzewa), 2008. Personal Communication.

[9] V. Chaoji, M. Al Hasan, S. Salem, and M. J. Zaki. Sparcl: an effective and efficient algorithm for mining arbitrary shape-based clusters. *Knowledge and Information Systems*, 2009.

[10] M. Cottrell, B. Hammer, A. Hasenfuss, and T. Villmann. Batch and median neural gas. *Neural Networks*, 19:762–771, 2006.

[11] M.G. Forero, F. Sroubek, and G. Cristobal. Identification of tuberculosis bacteria based on shape and color. *Real-time Imaging*, 10(4):251–262, 2004.

[12] I. Guyon. *Feature Extraction. Foundations and Applications.* Berlin, Springer, 2006.

[13] T. Hastie and W. Stuetzle. Principal curves. *J. Am. Stat. Assn.*, 84:502–516, 1989.

[14] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer, New York, 2001.

[3] Acknowledgment: We like to thank Markus Kostrzewa, Thomas Meier, Stefan Klepel for useful discussions. We appreciate the very helpful review work of reviewer one and three. Frank-M. Schleif was funded by the Federal Ministry of Education and Research under FZ:0313833 during this project.

[15] K. Hollemeyer, W. Altmeyer, E. Heinzle, and C. Pitra. Species identification of oetzis clothing with matrix-assisted laser desorption/ionization time-of-flight mass spectrometry based based on peptide pattern similarities of hair digests. *Rapid Comm. in Mass Spec.*, 22:2751–2767, 2008.

[16] S.-Y. Hsieh, C.-L. Tseng, and Y.-S. Lee. Highly efficient classification and identification of human pathogenic bacteria by MALDI-TOF-MS. *Molecular and Cellular Proteomics*, 7(2):448–456, 2008.

[17] A. Hu, A.A. Lo, C.T. Chen, K.C. Lin, and Y.P. Ho. Identifying bacterial species using CE-MS and SEQUEST with an empirical scoring function. *Electrophoresis*, 28(9):1387–92, 2007.

[18] A. Kalousis, J. Prados, and M. Hilario. Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowledge and Information Systems*, 12(1):95–116, 2007.

[19] C.J. Keys, D.J. Dare, H. Sutton, G. Wells, M. Lunt, T. McKenna, and M. McDowall H.N. Shah. Compilation of a MALDI-TOF mass spectral database for the rapid screening and characterisation of bacteria implicated in human infectious diseases. *Infect Genet Evol.*, 4(3):221–42, 2004.

[20] J. Khatun, E. Hamlett, and M. C. Giddings. Incorporating sequence information into the scoring function: a hidden markov model for improved peptide identification. *Bioinformatics*, 24(5):674–681, 2008.

[21] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 1995. (Second Extended Edition 1997).

[22] M. Kostrzewa. Efficieny of MS+BioTyper based bacteria identification for the clinical market. Personal Communication, 2009.

[23] H.J. Kushner and D.S. Clark. *Stochastic Appproximation Methods for Constrained and Unconstrained Systems*. Springer-Verlag, New York, 1978.

[24] K. Labusch, E. Barth, and T. Martinetz. Learning data representations with sparse coding neural gas. In M. Verleysen, editor, *Proceedings of the European Symposium on Artificial Neural Networks ESANN*, pages 233–238. d-side publications, 2008.

[25] K. Labusch, E. Barth, and T. Martinetz. Sparse coding neural gas: Learning of overcomplete data representations. *Neurocomputing*, 72:1547–1555, 2009.

[26] D. C. Liebler. *Introduction to Proteomics*. Humana Press, 2002.

[27] T. M. Martinetz, Stanislav G. Berkovich, and Klaus J. Schulten. 'Neural-gas' network for vector quantization and its application to time-series prediction. *IEEE Trans. on Neural Networks*, 4(4):558–569, 1993.

[28] Mathworks. MATLAB statistics-toolbox. http://www.mathworks.com (last visited 08.05.2009), 2008.

[29] M. F. Mazzeo, A. Sorrentino, M. Gaita, G. Cacace, M. Di Stasio, A. Facchiano, G. Comi, A. Malorni, and R. A. Siciliano. Matrix-assisted laser desorption ionization-time of flight mass spectrometry for the discrimination of food-borne microorganisms. *Applied and Environmental Microbiology*, 72(2):1180–1189, 2006.

[30] E. Oja. Neural networks, principle components and subspaces. *International Journal of Neural Systems*, 1:61–68, 1989.

[31] B.A. Olshausen and D.J. Finch. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[32] J. Pakkanen, J. Iivarinen, and E. Oja. The evolving tree—a novel self-organizing network for data analysis. *Neural Process. Lett.*, 20(3):199–211, 2004.

[33] D. Pfitzner, R. Leibbrandt, and D. Powers. Characterization and evaluation of similarity measures for pairs of clusterings. *Knowledge and Information Systems*, 19(3):361–394, 2009.

[34] S. Saha and S. Bandyopadhyay. A new multiobjective clustering technique based on the concepts of stability and symmetry. *Knowledge and Information Systems*, 2009.

[35] F.-M. Schleif, M. Lindemann, P. Maass, M. Diaz, J. Decker, T. Elssner, M. Kuhn, and H. Thiele. Support vector classification of proteomic profile spectra based on feature extraction with the bi-orthogonal discrete wavelet transform. *Computing and Visualization in Science*, pages DOI: 10.1007/s00791–008–0087–z, 2008.

[36] F.-M. Schleif, T. Villmann, M. Kostrzewa, B. Hammer, and A. Gammerman. Cancer informatics by prototype networks in mass spectrometry. *Artificial Intelligence in Medicine*, page PMID:18778925, 2008.

[37] O. Schmid, G. Ball, L. Lancashire R. Culak, and H. Shah. New approaches to identification of bacterial pathogens by surface enhanced laser desorption/ionization time of flight mass spectrometry in concert with artificial neural networks, with special reference to neisseria gonorrhoeae. *J Med Microbiol.*, 54:1205–11, 2005.

[38] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, 2004.

[39] S. Simmuteit. Effizientes Retrieval aus Massenspektrometriedatenbanken, Diplomarbeit, Technische Universität Clausthal. February 2008.

[40] A. Ultsch and H. P. Siemon. Kohonen's self organizing feature maps for exploratory data analysis. In *Proc. INNC'90, Int. Neural Network Conf.*, pages 305–308, Dordrecht, Netherlands, 1990. Kluwer.

[41] N. Valentine, S. Wunschel, D. Wunschel, C. Petersen, and K. Wahl. Effect of culture conditions on microorganism identification by matrix-assisted laser desorption ionization mass spectrometry. *Appl Environ Microbiol*, 71(1):58–64, 2005.

[42] T. Villmann and J.-C. Claussen. Magnification control in self-organizing maps and neural gas. *Neural Computation*, 18(2):446–469, February 2006.

[43] T. Villmann, R. Der, M. Herrmann, and T. Martinetz. Topology Preservation in Self–Organizing Feature Maps: Exact Definition and Measurement. *IEEE Transactions on Neural Networks*, 8(2):256–266, 1997.

[44] T. Villmann, F.-M. Schleif, B. Hammer, and M. Kostrzewa. Exploration of mass-spectrometric data in clinical proteomics using learning vector quantization methods. *Briefings in Bioinformatics*, 9(2):129–143, 2008.

[45] J.G. Wilkes, K.L. Glover, and M. Holcomb. Defining and using microbial spectral databases. *J Am Soc Mass Spectrom.*, 13(7):875–87, 2002.

[46] Z. Zhang, G.W. Jackson, G.E Fox, and R.C. Willson. Microbial identification by mass cataloging. *BMC Bioinformatics*, 7:117, 2006.

Author Biographies

Stephan Simmuteit received a diploma degree in computer science from the Clausthal University of Technology, Germany. In 2008 he joined the Computational Intelligence Group at the University of Leipzig where he is currently studying for his Ph.D. His research interests include machine learning, self-organization and high-dimensional data analysis.

Frank-Michael Schleif received his Ph.D. in Computer Science in 2006 from Clausthal-University of Technology, Germany. From 2004-2006 he was a software developer and research scientist at Bruker Biosciences. Since 2006, he is a research scientist at the University of Leipzig in the project *metastem* and since 2009 group leader of the Computational Intelligence Group at the Medical Department of the Leipzig University. His research activities focus on machine learning methods, spectral processing, statistical data analysis and algorithm development.

Barbara Hammer received her Ph.D. in Computer Science in 1995 and her venia legendi in Computer Science in 2003, both from the University of Osnabrueck, Germany. From 2000-2004, she was leader of the junior research group 'Learning with Neural Methods in structured Domains' before accepting an offer as professor for theoretical computer science at Clausthal University of Technology. Her areas of expertise include clustering and classification, recurrent and structure processing neural network, self-organization, statistical learning theory, and bioinformatics.

Thomas Villmann is Professor for Applied Mathematics and Computational Intelligence at the University of Applied Sciences Mittweida (Germany) and leads the research group Computational Intelligence. He holds a diploma degree in mathematics and a Ph.D. as well as the venia legendi in Computer Science. His research areas comprise the theory of prototype-based vector quantization and classification, self-organizing neural networks and machine learning approaches as well as the utilization of non-standard metric and metric adaptation for optimum classification. Further he is involved in respective applications in pattern recognition, medical data analysis, bioinformatics, mass-spectrometry data analysis and satellite remote sensing. Several research stays have taken him to Belgium, France, the Netherlands, and the USA. He is a founding member of the German chapter of the European Neural Network Society (GNNS). He is editor of the Machine Learning Reports and member of the editorial board of the Neural Processing Letters.

3.4 Genetic algorithm for shift-uncertainty correction in 1-D NMR based metabolite identifications and quantifications

The article *Genetic algorithm for shift-uncertainty correction in 1-D NMR based metabolite identifications and quantifications* by F.-M.**Schleif**, T. Riemer, U. Börner, L. Schnapka-Hille and M. Cross appeared in **Bioinformatics** 27 (4), p. 524–533, 2011. In the article the Extended Targeting Profiling (ETP) is proposed, a method for the (semi-) automatic identification and quantification of metabolites in nuclear magnetic resonance (NMR) spectra. The method employs a specific genetic algorithm approach coupled with a functional distance measure to optimize the theoretically derived spin-system parameter model with respect to the observed data. The optimized parameters can be used to generate prototypical metabolite spectra. This article is based on an interdisciplinary work between the chemist T. Riemer, responsible, mainly for the NMR equipment, the biochemist U. Börner, responsible for the NMR wet-lab experiments and two biologists L. Schnapka-Hille and M. Cross, responsible for the biological modeling and cell experiments. In this article I developed the preprocessing and optimization methods for the analysis of the spectra. The algorithm has been developed by myself with some support by T. Riemer to incorporate the NMR simulation environment used in ETP. I conducted all non-wet lab experiments and wrote the majority of the paper. My co-author T. Riemer helped to integrated the gamma simulation environment and wrote the technical part of the NMR related section. U. Börner did the wet-lab experiments for the cell extracts and artificial samples, she also wrote the corresponding section to describe the experimental designs. L. Schapka-Hille did the experiments with the cell lines and M. Cross supervised the lab experiments. All authors discussed the general article.

Genetic algorithm for shift-uncertainty correction in 1-D NMR based metabolite identifications and quantifications

F.-M. Schleif [1], T. Riemer [2], U. Börner[2,3], L. Schnapka-Hille[3], M. Cross[3]
[1]Univ. of Bielefeld, Dept. of CS, Bielefeld, Germany
[2]Univ. of Leipzig, Dept. of Med. Phys. and Biophys.,Leipzig, Germany
[3]Univ. of Leipzig, Dept. of Hematology, Leipzig, Germany

August 8, 2012

Abstract

1 Motivation:

The analysis of metabolic processes is becoming increasingly important to our understanding of complex biological systems and disease states. Nuclear magnetic resonance spectroscopy (NMR) is a particularly relevant technology in this respect, since the NMR signals provide a quantitative measure of metabolite concentrations. However, due to the complexity of the spectra typical of biological samples, the demands of clinical and high throughput analysis will only be fully met by a system capable of reliable, automatic processing of the spectra. An initial step in this direction has been taken by *Targeted Profiling* (TP), employing a set of known and predicted metabolite signatures fitted against the signal. However, an accurate fitting procedure for ^{1}H NMR data is complicated by shift uncertainties in the peak systems caused by measurement imperfections. These uncertainties have a large impact on the accuracy of identification and quantification and currently require compensation by very time consuming manual interactions. Here, we present an approach, termed *Extended Targeted Profiling* (ETP), that estimates shift uncertainties based on a genetic algorithm (GA) combined with a least squares optimization (LSQO). The estimated shifts are used to correct the known metabolite signatures leading to significantly improved identification and quantification. In this way, use of the automated system significantly reduces the effort normally associated with manual processing and paves the way for reliable, high throughput analysis of complex NMR spectra.

2 Results:

The results indicate that using simultaneous shift uncertainty correction and least squares fitting significantly improves the identification and quantification results for ^{1}H NMR data in comparison to the standard targeted profiling approach and compares favorably with the results obtained by manual expert analysis. Preservation of the functional structure of the NMR spectra makes this approach more realistic than simple binning strategies.

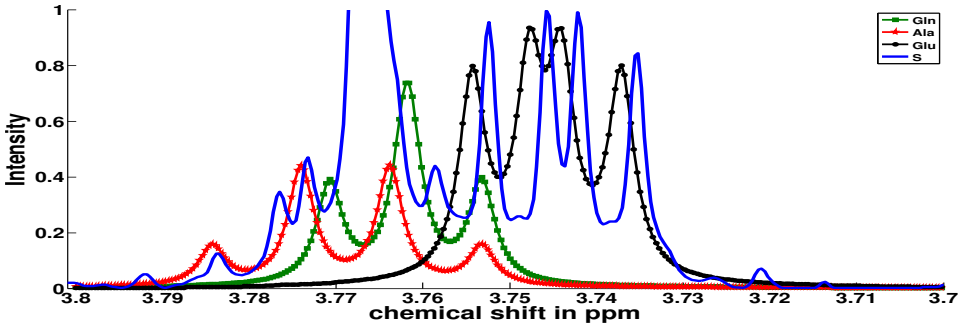


Figure 1: Overlapping effect in a ^{1}H NMR spectrum of multiple metabolites. It can clearly be seen, that the assumption of the Lorentzian fails to provide an accurate approximation in some regions. This can lead to incorrect estimates of target heights and hence wrong concentration estimates.

3 Availability:

The simulation descriptions and scripts employed are available under: http://139.18.218.40/metastemwww/bioinf/bioinf_suppl_nmr_ga_opt_schleif_et_al.tgz

4 Contact:

schleif@informatik.uni-leipzig.de

5 Introduction

The quantitative profiling of metabolites and the mathematical modeling of metabolic networks is set to make a major contribution to our understanding of complex biological systems, including the processes underlying development and tissue homeostasis ([29]). The most commonly used methods for metabolite detection are mass spectrometry (MS) and nuclear magnetic resonance spectroscopy (NMR). While each has its specific advantages, the inherently quantitative nature of NMR makes it most attractive for providing data for the development of mathematical models. However, the current challenge is to extract reliably quantitative data from experimental spectra which are often complex and subject to background variability. Here we focus on the exact extraction of metabolite information from ^{1}H NMR measurements. The general strategy involves pre-processing steps such as phase- and baseline correction, smoothing and data reduction ([31, 3]), followed by the identification of distinct metabolite signatures in the signal and the estimation of metabolite concentrations with respect to the original biological samples. Details of the basic pre-processing used in this work are provided in ([21, 22]). A number of approaches have been reported to help in the subsequent identification and quantification of individual metabolites from preprocessed data ([1, 32, 33, 30]). However, none of the methods currently available can be applied in the reliable, automated fashion necessary for the high-throughput processing of complex biological samples ([17, 15]). As an initial step towards automatic processing, *targeted profiling* (TP) ([30]), employs a set of known and predicted metabolite signatures (targets) fitted against the signal. However, an accurate fitting procedure for ^{1}H NMR data is complicated by small but significant shift uncertainties in the peak systems, caused by even minor variations in parameters such as temperature and pH ([4]).

These uncertainties have a large impact on the accuracy of identification and quantification and currently need to be compensated by very time consuming manual interactions. Independent correction of the shift followed by fitting of the corrected target descriptions against the signals is not generally feasible because of the strong overlaps typical of ^{1}H NMR spectra.

Generic methods for the compensation of peak shifts are typically based on a specific or average reference signal taken from the data ([7]). If such a reference is available, then the NMR spectra are locally aligned to it such that the final set of spectra is reasonable aligned and corresponding peaks match. The used optimization techniques commonly employed include partial least squares approaches ([28]), genetic algorithms ([6]) and procedures based on the fourier transformation ([19]). This type of alignment problem is relevant not only to NMR but also to other data, including mass spectrometry ([18, 20]). While the proposed approaches are promising and reasonable fast, they assume the availability of a reference spectrum to be used as the objective goal. Sometimes it is merely assumed that a set of common reference peaks is available so that an alignment function can be estimated based on these data ([20]). However, this is often not realistic and in the setting considered here we do not assume the existence of a (global) reference spectrum. Furthermore, even for the aligned spectra one can not ensure that the peaks are aligned to their *true* position, only they are aligned to one another. If the chosen reference is not an undisturbed signal then there is no guarantee that the aligned spectra show correct ppm or mass positions for the peaks. In the case of metabolic profiling, this leaves the problem of correct identification and quantification of the metabolites in a spectrum with potential peak shifts. Our approach focuses on this special problem. The prior mentioned alignment methods can be used as a potential preprocessing only if the analyzed spectra are reasonable similar, as it should be the case for replicates. In this case it is possible to align the spectra first before using the approach, presented below.

The targets consist of a set of parametrized peak models showing uncertainties in their positions with respect to a true measurement, as described in more detail below. A typical NMR signal from a biological sample containing a variety of targets contains around 100 erroneous shift parameters. Local shift uncertainties need to be corrected within a given tolerance for all these parameters and often within the context of overlapping targets. Furthermore, NMR data show very spiked peaks so that both the correct peak positions and accurate target height estimates are decisive to the accuracy of metabolite concentration estimates. This makes a complete evaluation of all possible solutions unfeasible and the problem is ill posed.

We present here an approach designed to improve this situation by semi-automatic analysis of the spectra such that only minor, simple interaction steps are necessary to allow the processing of large data sets. We developed an approach estimating shift uncertainties based on a genetic algorithm (GA) ([8, 16]) combined with a least squares optimization (LSQO) ([5]). Genetic algorithms are known to be very effective in finding local optimal solutions for ill-posed problems and have already been applied to spectroscopic data ([10, 9]). The estimated shifts are used to correct the known metabolite signatures, leading to significantly improved identification and quantification results. The shift uncertainties are generally corrected with sufficient accuracy that little or no subsequent manual interaction is necessary to generate the final quantifications. The method has been tested on a range of NMR spectra obtained from cell culture experiments. We have evaluated the models obtained in comparison to a standard targeted profiling approach as well as to the defacto standard of a careful manual analysis. We have also studied the observed shift uncertainties with respect to their influence on the concentration estimates during the multiple steps of the GA.

6 Approach and Methods

6.1 NMR Spectroscopy

All ^{1}H NMR-spectra were acquired on an AVANCE 700 MHz NMR-spectrometer (Bruker, Rheinstetten, D) equipped with a 5 mm cryo-probe. A pulse acquire sequence was used with 512 accumulations, 65536 complex points, 8389.2 Hz sweep width corresponding to 11.982 ppm on the chemical shift axis (0.002 ppm , 0.13 Hz nominal spectral resolution, respectively) and a repetition time of 20 seconds (> five times the T1 of the reference and metabolites) ensuring fully relaxed, quantifiable signals. NMR samples were prepared by re-suspending lyophilised cell extracts in $500\mu l$ D_2O (99.9 atom %, Sigma Aldrich, Steinheim, D) potassium phosphate-buffer (0.05M, pH 7.4) containing a known concentration ($60-120\mu$M) of $2,2$'dimethylsilapentane-5-sulfonate (DSS, 99.0%, Fluka, Taufkirchen, Germany) as a reference for chemical shift and quantification. Each extract was then mixed vigorously by vortexing and centrifuged for 4 min at 10.000g. The supernatants (approx. 500μl) were transferred to 5 mm NMR-tubes (Wilmad, Vineland NJ USA). All samples were subject to NMR analysis at 298 K within 12 h.

6.2 Data pre-processing

We focus on the analysis of ^{1}H liquid NMR spectra obtained from extracts of cultured stem/progenitor cells, detailed subsequently. Each spectrum was preprocessed using in-house Matlab ([14]) routines. Spectra were phased, baseline corrected and referenced using DSS as a chemical shift and shape indicator (CSI) [1]. Furthermore, the region around ($4.5-5.9ppm$) was set to zero for each spectrum to remove the water resonance contributions. Further details on the basic pre-processing are given in ([21, 22]).

6.3 Data set description

We employed a set of 6 NMR spectra from cells cultured under a range of conditions to provide biologically realistic degrees of sample complexity and variation. The expected metabolites in the signal (subsequently referenced as targets) were: Alanine - (Ala), Asparagine - (Asn), Aspartate - (Asp), Citric Acid - (Cit), Cysteine - (Cys), Glutamate - (Glu), Glutamine - (Gln), Glycine - (Gly), Histidine - (His), Iso-Leucine - (Ile), Lactate - (Lac), Leucine - (Leu), Malate - (Mal), Methionine - (Meth), Myo-Inositol - (Myo), Phenyl-Alanine - (Phe), Proline - (Pro), Pyruvate - (Pyr), Serine - (Ser), Succinate - (Succ), Threonine - (Thr), Tryptophan - (Trp), Tyrosine - (Tyr), Valine - (Val), Fumarate - (Fum) and DSS as the standard reference. The signal is also expected to contain some unspecified metabolites.

The murine multipotent hematopoietic progenitor cell line FDCPmix (Factor Dependent Cells Paterson mixed potential) was grown in IMDM supplemented with 5 mM D-glucose, 2 mM L- glutamine, 1 mM sodium pyruvate, 20% horse serum and 10 u/ml IL-3. Six independent cultures were analysed, generated separately over a period of 18 months under the same culture conditions. The cells were maintained at 37 °C in 5% CO_2 in air at densities between 6×10^4 and 5×10^5 cells per ml by passaging every $2-3$ days. At the final passage, the cells were transferred to fresh medium and cultured for 3 days. Between 1×10^8 and 2×10^8 cells from each experiment were harvested by centrifugation and washed four times with ice cold phosphate buffered saline (PBS) to remove medium constituents. The cell pellets were shock frozen in liquid nitrogen and extracts prepared by addition of $800\mu l$ ice

[1] Other choices for the CSI e.g. trimethylsilyl propionate (TSP) are also possible. The ideal CSI is only one peak with no overlap to other peaks.

cold methanol:acetonitrile:water $1:1:1$ mixture. To ensure efficient cell disruption the cells were subjected to 2×1 minute bursts of ultrasound in an ice cold ultrasonic bath. The samples were then transferred to a $70\,^{\circ}\mathrm{C}$ water bath for 10 minutes to denature the proteins before being diluted $1:7$ with water and lyophilized.

Additionally we analyzed a set of 4 spectra of wet-lab mixtures of the 5 metabolites (Ile,Leu,Glu,Val,Meth) and DSS as a standard with known concentrations.

6.4 Manual NMR expert analysis

The metabolites of interest were first measured individually by NMR to provide reference-spectra. A known concentration of the metabolite (1 - 20 mM) together with DSS ($0.1 - 2$ mM) was prepared in $500\mu l$ buffered D_2O solute (see 2.1) and measured under the same conditions as those used for the cell extracts. This allowed the determination of all chemical shifts (σ) and coupling constants (J) of each signal-generating metabolite proton as a basis for the reliable identification of metabolites in subsequent experiments.

Metabolite identification and quantification was achieved using purpose-developed NMR software (NMRj,[23]) allowing for the interactive subtraction of a simulated from a measured NMR- spectrum. The chemical shifts and coupling constants from the simulation were carefully adjusted within a range of < 0.01 ppm to enable stringent fitting of the frequency pattern of the individual spin systems to the cell extract-spectrum. The criteria for an acceptable fit were firstly that all of the simulated peaks be present in the measured NMR-spectrum (i.e. identification of the metabolite) and secondly that the difference spectrum resulting from subtraction of the simulation from the measurement exhibited a smooth baseline at the position of metabolite frequencies. The latter step requires that the simulated signal is folded by a line broadening function that is as close as possible to that of the measured spectrum. This was achieved by using up to three exponential broadening functions, independent in amplitude, damping and frequency offset, for folding the simulated spectral time signal. Metabolite concentrations were calculated from the identified metabolite's NMR time-signal amplitude relative to the time signal amplitude of the known DSS reference concentration taking into account the relative number of contributing protons.

6.5 NMR and targeted profiling

High resolution ^{1}H NMR spectra consist of a large number of relevant signals. Metabolite signatures are represented in general by multiple narrow peaks located on top of a wide underlying complex baseline. The NMR signal $s(\nu)$ can be approximated as a super composition of Lorentzians ([11]), Gaussian functions or mixtures thereof. However, such assumptions are highly idealized. In practical measurements the line shape of the peaks is much more complex and inhomogeneous due to measurement imperfections. This poses multiple challenges in the analysis because almost all relevant signals in the NMR measurement show strong overlapping components. Without an appropriate model of the signal structure and line shape a deconvolution is extremely complicated. This is especially true for signal components at low concentrations which may otherwise be easily overlooked.

The TP approach ([30]) analyses metabolites by referencing to a set of known signatures. Taking some relatively strong assumptions concerning the line shape and knowledge about the structure of the targets, TP tries to identify and quantify these target metabolites in the complex NMR spectrum.

The TP approach assumes an almost perfect knowledge of the peak or line shape, which is typically modeled as a Lorentzian or a Gaussian function. It is also assumed, that the number of candidate signatures in the mixture $s(\nu)$ is small and restricted to a specific

subset of known metabolites, the targets. Furthermore, it is assumed that for all targets, their peak sequence, i.e. the signal signature defined by the position and height of the peaks, is known perfectly beforehand. In practice it is often very difficult to provide such a description analytically for complex mixtures with extensive overlaps. For this reason the peak system is constructed (manually) by adding appropriate peaks at the correct ppm position and height. The targets are subsequently fitted against the measurement.

TP is being adopted as a standard technique in metabolite analysis and has already been employed in a number of studies see e.g. ([27, 26, 25]). While TP has been found to be very effective in a range of applications it remains suboptimal in many cases: (1) Due to variations in the measurement conditions (e.g. temperature, pH) the position of the g_i in a target (groups of peaks) may shift in a non-linear manner. (2) A specific line shape has to be chosen for the fitting of the candidate targets against the signal. Since the actual line shape may deviate from the chosen forms, this assumption can lead to further problems especially for strongly overlapping signals as depicted in Figure 1. (3) The simple fit of individual targets against the signal $s(\nu)$ may fail for strongly overlapping structures, while the use of lower constraints on the fitting commonly leads to incorrect identifications of targets. In the later case it can happen that lines are fitted into regions without signal.

The TP approach also lacks the formal and mathematical derivation and modeling basis which would simplify adaptations, for instance to accommodate moderate changes in the device settings such as alternative measurement frequencies, or to incorporate alternative peak shape models.

In the following section we formalize targeted profiling and detail our extension thereof. We provide an appropriate mathematical modeling for the fitting and parameter estimation approach, taking the functional characteristic of the measurements into account.

7 Extended Targeted Profiling

An arbitrary metabolite may formally be given by a *functional description* $f(\nu)$ for a target signal as $f(\nu) = \sum_j^G g_j(\nu)$ with $g_j(\nu)$ as a peak pattern or a function of delta functions with non-zero entries only on the appropriate peak positions as detailed below and G as the number of such peak patterns. Using the TP approach $f(\nu)$ may be folded with an appropriate line shape e.g. a Gaussian. A reconstruction of alanine using the functional description is given in Figure 2.

An alternative compact description of a target e.g. alanine is given by its ^{1}H NMR *spin system classification* as A_3X spin system (see e.g. [13]), with the associated values for the chemical shifts of $\sigma(H_A) = 1.46$ ppm, $\sigma(H_X) = 3.76$ ppm and an A-X coupling constant of $J_{AX} = 7.2$ Hz see Figure 3.

Using the above spin system classification, we can employ a NMR simulation environment ([24]) to simulate the alanine spectrum whilst taking the physical properties of our measurement system (such as device frequency) into account.

This simulation yields transition tables providing information on the peak positions and heights of each peak for the target. A transition table for L-alanine is shown in Table 1.

From this line spectrum we can generate a profile spectrum, similar to a true measurement by folding the line spectrum with an assumed line shape, leading to our functional description $f(\nu)$ of a given target (see Figure 2). Taking this approach we can model the,

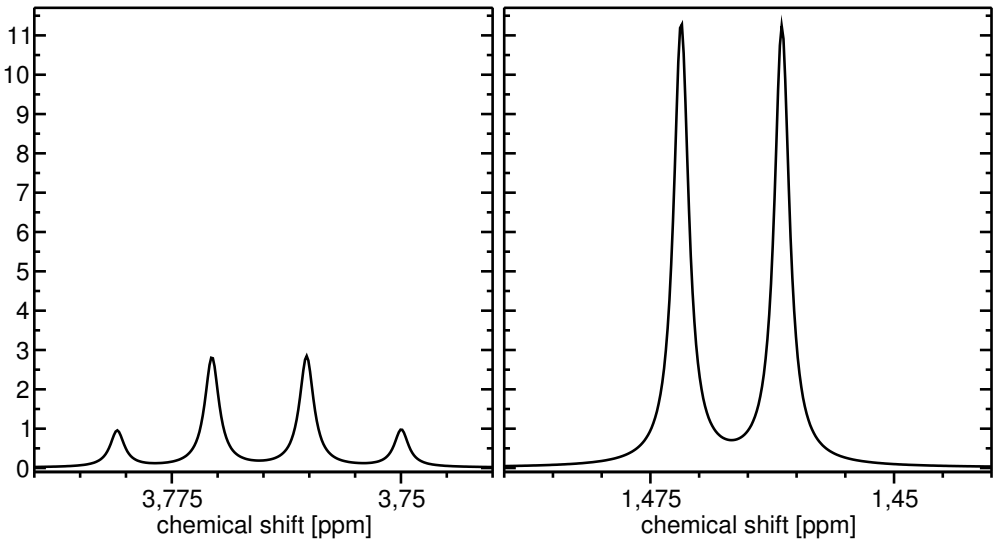


Figure 2: Reconstruction of L-alanine using the functional description. The x-axis is given in ppm and the y-axis shows the intensities. (a): the quartet generated by the H_x proton with a shift parameter $\sigma(H_x)$ and (b): the doublet caused by the three magnetically equivalent H_A protons with shift parameter $\sigma(H_A)$.

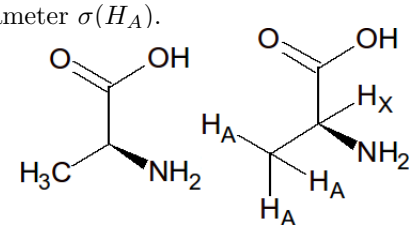


Figure 3: Structure of L-alanine (left) and in A_3X notation (right).

phased and baseline corrected signal $s(\nu)$ as

$$s(\nu) = (\sum_{j}^{J} \alpha_j f_j(\nu - o)) + \epsilon \quad (1)$$

$$f_j(\nu) = \sum_{i}^{G_j} g_i(\nu - \Delta_i) \quad (2)$$

$$g_i(\nu) = \sum_{k}^{K_{j,i}} \Theta_k(\nu) \otimes \wp(\nu) \quad (3)$$

$$\wp = \text{e.g.}\exp(\ldots) \quad \text{line shape} \quad (4)$$

We employ a non-negative Least Squares Fit over all J identified targets $f_j(\nu)$ using the functional description and the subsequently generated peak information. Thereby o represents a global shift which can be compensated by a reference shift correction and ϵ represents noise. The target f_j can be approximated as a super composition of its component functions or peak groups g_i defined by the number G_j of chemical shifts in the molecule's spin system.

A small local shift $-\gamma \leq \Delta_i \leq +\gamma$ typically within a range of $|\gamma| \leq 0.005$ ppm can

Index	PPM	Intensity	Group index	A_3X
1	1.4596	11.2246	1	A_3
2	1.4700	11.2752	1	A_3
3	3.7499	0.9438	2	X
4	3.7603	2.8187	2	X
5	3.7706	2.8061	2	X
6	3.7809	0.9311	2	X

Table 1: Transition table providing the information for a line spectrum reconstruction of L-alanine. The table was generated using standard settings for a 700.153 MHz NMR system ^{1}H channel as specified before.

be expected for each peak group. Each component $\Theta_k(\nu)$ of $g_i(\nu)$ can be considered as a delta function, contributing to a line spectrum with non vanishing amplitude for one peak position only. We denote such a single position ν as $\nu_{j,i,k}$ to specify peak k caused by group i in metabolite j. K is the multiplicity of a component function g_i. The origin of the chemical shift group components $\Theta_k(\nu)$ lies in the spin-spin interaction characterized by the scalar coupling constant J_{AX} and can be deduced from the quantum mechanical calculations for the spin system parameters describing the target metabolite. Subsequently this line spectrum is folded $\otimes$ by a line shape function $\wp$ to mimic the line shape of the real measurement. In the following we will use G for G_j and K for $K_{j,i}$ if the indices are known from the context. In NMR the position of the g_i are known as chemical shifts. The estimates of these shift positions need to be as accurate as possible and are the main error-source in the TP approach.

An accurate peak shape estimate is the key to an appropriate subtraction of signal components from $s(\nu)$ in order to reveal potentially hidden components. We approach this issue by taking the shape of the DSS reference signal added to the sample, as a template for $\wp$. This shape is used to estimate the expected peak width present in the signal.

To tackle the shift-uncertainty problem, we estimate values for the disturbances Δ shown in Eq. (2) and present an initial solution to optimize the g_i positions in potential targets using a grid search strategy. This approach leads to a general improvement in position estimates for the *true* chemical shifts of the sub-patterns g_i of potential targets f_i and hence to more accurate identification and quantification estimates as shown below.

Whereas standard TP identifies signatures in NMR mixtures by employing known database references of (manually) specified peak patterns the Extended Targeted Profiling approach (ETP) described here modifies this concept by modeling the targets based on their theoretical spin-system model (see ([24]). This model provides the peak information (transition tables). The physical model easily deals with measurement variables such as different device frequencies and is known to provide very accurate peak lists. The parameters of the targets are optimized with respect to the measurements at hand. Each target description T (generating a signal $f_j(\nu)$) is characterized by a set of spin-system descriptors $T_d \in S$. S describes the theoretical aspects of the spin system of T and can be used in combination with a model of the measurement system (NMR system) to simulate the spectrum f_j for T. A spectrum representation of T can be divided into multiple parts, one for each spin-system descriptor T_d, known as the peak group (g). A peak group may consist of multiple or single peaks and is potentially overlapping. For each group a potential (limited) shifting uncertainty Δ_i can be expected. New targets can be added to the ETP approach very easily by specifying the spin-system model, outlined above, based either on knowledge available in the literature or by own measurements of the pure target substance under the previously defined measurement conditions. In the latter case the obtained spectrum is analyzed manually to define

the spin-system model. Hereby an NMR expert constructs a spin-system model such that the reconstructed spectrum, based on this model, fits best to the observed data. The three steps of ETP required to obtain an optimized fit based on this new encoding strategy are detailed below.

7.1 Line representation of a NMR spectrum

NMR spectra can be described by means of a set of overlapping peaks, which provides a compact representation of the signal and can also reveal quickly whether or not an expected target is likely to be present in $s(\nu)$, since all simulated target peaks must also be present in the peak list of $s(\nu)$. The peak picking process is rather complicated, and a number of heuristic approaches have been proposed to improve the situation ([11, 2]). Here we focus on a simple parametric hill-climbing approach ([22]). We further assume that for each measurement a known CSI signal is available, in our case this is DSS. This signal has a known position of 0 ppm, which can be used to compensate the global shift offset of the spectrum. We look for a maximum within a window of $0.05ppm$ at the expected CSI position. From this position we then go down (to lower intensities) on the left and the right flank of the peak as long as the signal is a descending monotone. The peak is then truncated at a predefined maximal width. The center position and the peak width at half maximum (PWHM) are then calculated for this peak. The PWHM is used as a rough estimate of the peak width. Due to effects such as imperfect phasing, shimming or baseline correction, a direct inverse deconvolution of $s(\nu)$ with the CSI reference is not generally feasible. Instead, we employ a hill-climbing algorithm and look above a predefined threshold (the expected noise level) through the whole signal for local maxima, whose flanks are sufficiently steep and for which the obtained peak has a sufficient width. By application of this algorithm we obtain a list of peaks in a spectrum. This list is subtracted from $s(\nu)$ and the algorithm is repeated until no further peaks are detected. This approach can also resolve peaks in an overlap, although not in every case. Alternatively, the strategy described in ([11]) can be used with an underlying Lorentzian support, the particular peak picking algorithm is not of much relevance here as long as it discovers the peaks in the spectrum to a sufficient degree of accuracy. The list of peaks is subsequently denoted as $\mathcal{P}$. These peak lists are compared to those of the potential targets. If a sufficient number of peaks (e.g. 30%) in a target can be matched within a tolerance of $0.01ppm$ to the peaks $\mathcal{P}$ we consider the target to be identified and proceed with the analysis steps for this target. We now have the target as a functional line spectrum $f_j(\nu)$ with $\wp$ as the fitted line function.

7.2 Genetic algorithm for shift uncertainty estimation

A major feature of our approach is the shift uncertainty correction performed by means of a Genetic Algorithm (GA). The genetic algorithm software was written in-house in Matlab running on an Intel Xeon multiprocessor system with 8 3.20 GHz processors and 16 GB memory using the parallel processing, signal processing and optimization toolbox with Matlab 2008b. We made use of the GA implementation in the optimization toolbox but replaced some of the core methods with our own purpose-developed implementations. Specifically, we replaced the methods used to generate the initial population and the mutation function and provided a specific fitness function as described below. The basic algorithm and parameters for the GA are shown in Table 2 and the overall workflow is depicted in Figure 4. Briefly, we generate a large number of chromosomes P, each chromosome has the same length $Z = \sum_j^J G_j$, equal to the number of groups over all analyzed targets, and each of which contains the currently estimated, or randomly determined shift values for the Δ_i. These Δ_i are

Parameter	Description	Value
C	single chromosome	$c \in \mathbb{R}^Z$ $c_i \in [\text{-PPM-E,PPM-E}]$
M	set of chromosomes	$M = \{C_1, \ldots, C_P\}$
K	Number of generations	200
P	Number of chromosomes	900; $\|M\|$
p_i	permutation probability	0.1
Z	Length of the chromosomes	$\sum G_j$
PPM-E	PPM uncertainty of the Δ_i	0.01
SR	Down-sampling rate	4
d	distance measure	{0,1}

Table 2: Basic parameters of the genetic algorithm. The distance measure is either euclidean - 0, or a functional distance - 1.

optimized by the GA. Furthermore the smallest shift is limited by the ppm-axis resolution. We have found that a reduction of the original spectrum to $16k$ points corresponding to 0.5 Hz spectral resolution, is possible, whilst maintaining structural information of sufficient quality for the subsequent identification task. In this case up to 25 valid shift positions are possible for a given shift uncertainty of $\pm 0.01 ppm$ [2].

7.2.1 Fitness function and evaluation measures

The fitness function is the core element of the GA and is evaluated for each single chromosome separately. It consists of three procedures: (1) The spin-system classifications of all identified metabolites are used to generate the corresponding spectral representation. Thereby, the shifts given by the chromosome are applied to the corresponding groups g_i, and the reconstructions folded with the prior estimated line shape. We denote the matrix of all reconstructions $f_j^*(\nu)$, given as row vectors, as the matrix R

$$R = \begin{pmatrix} f_1^*(\nu) \\ \ldots \\ f_J^*(\nu) \end{pmatrix} \quad R' = \begin{pmatrix} f_1^*(\bowtie) \\ \ldots \\ f_J^*(\bowtie) \end{pmatrix}.$$

(2) These reconstructions are reduced to a range representation such that a compact form of R denoted as R' is obtained. In R' not all values for ν are used but only a limited set of ν in form of potentially overlapping range vectors $\bowtie_l = [\nu_l - (2 \cdot PPM - E) : \nu_l + (2 \cdot PPM - E)]$ with l as an index of a peak positions in Υ.

We collect all peak center positions of the metabolites denoted as $\Upsilon = \{\Upsilon_1, \ldots, \Upsilon_J\}$ with $\Upsilon_j = \{\nu_{j,i,k}\}_{i,k}^{G \times K}$ and $\nu_l \in \Upsilon$. Here we also incorporate the Δ_i and take the peak positions from the transition tables extended to a range of twice the assumed ppm-uncertainty PPM-E for each peak. A reduced test spectrum $s'(\bowtie)$ is constructed, accordingly.

(3) The matrix R' and the vector $s'(\bowtie)$ are subsequently used in the LSQO, as the third step, to calculate the α_i for all targets. The reduction to a range based representation is useful to avoid very large and extremely sparse matrices, which would complicate the subsequent α estimations. This step has no detrimental effects on the $\alpha-$estimates.

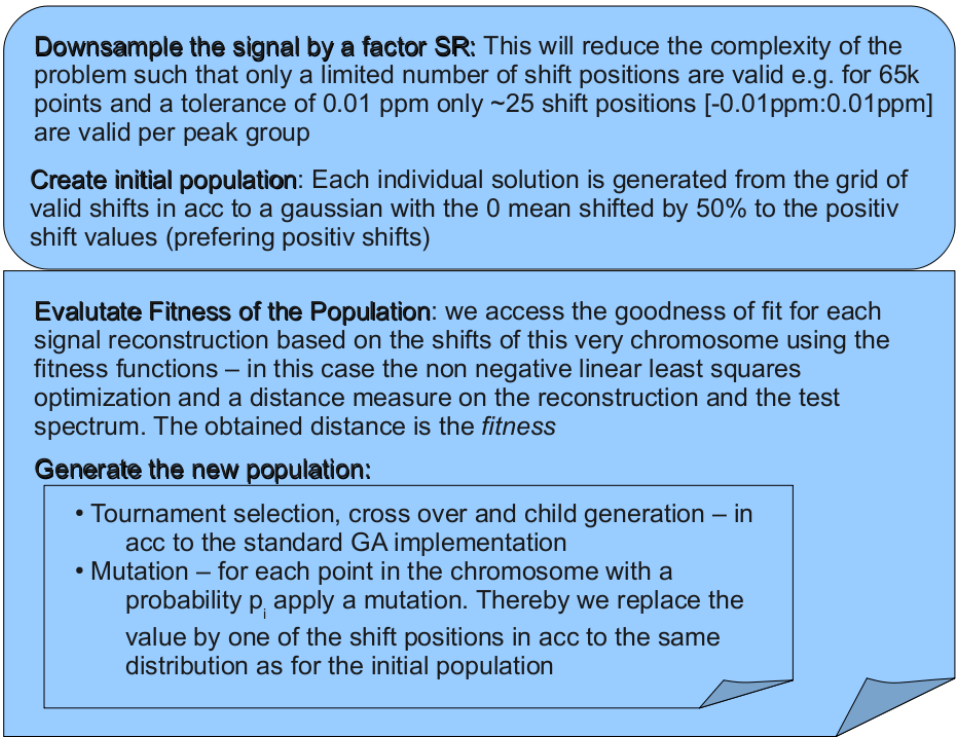
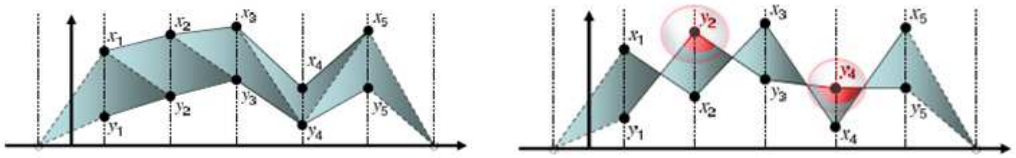


Figure 4: Workflow of the genetic algorithm used to obtain optimal Δ_i. The first (outer) folded corner is repeated until K generations are analyzed or one of the alternative standard stopping criteria is met. The inner folded corner is repeated until a new population of the same size as before is generated.



(a) Two functions: Euc = L^p-norm (b) Two functions: Euc $\neq$ L^p-norm

Figure 5: Illustration of the L^p-norm. Plot (a) indicates the case in which the distance between two functions is equal, both for Euclidean or L^p-norm. In plot (b) parts of the functions are interchanging (crossing). The distance using Euc is still the same as in plot (a) but for the L^p-norm the distance is changed, giving a more realistic measure of the distance of the two functions.

	Ile			Leu			Val		
	EXP	TP	ETP	EXP	TP	ETP	EXP	TP	ETP
M-DS$_1$	113.23	31.07	62.32	32.35	17.29	23.53	36.27	17.88	18.59
M-DS$_1$-C		72.64			37.77			21.85	
M-DS$_2$	36.32	26.00	73.98	62.96	15.00	21.43	32.78	15.98	28.52
M-DS$_2$-C		60.54			50.37			43.71	
M-DS$_3$	51.92	19.62	27.10	60.30	13.52	21.44	51.45	16.29	32.00
M-DS$_3$-C		36.32			62.96			32.78	
M-DS$_4$	57.98	21.27	36.01	58.85	16.55	25.94	34.33	14.50	11.46
M-DS$_4$-C		48.43			62.96			21.85	
$\otimes$		0.63	**0.46**		0.74	**0.60**		0.55	**0.43**

Table 3: Concentrations of metabolites in the synthetic wet-lab study. The weighted sample concentration is given in the -C rows each. The estimate of the expert EXP, TP and ETP are given in the columns. All concentrations are given in μ mol. Considering the median relative error ($\otimes$) of the concentration estimates the ETP approach is best in all cases. In a case by case comparison ETP is almost always the best, with three exceptions $\{(Spec_2, Val), (Spec_4, Glu), (Spec_1, Ile)\}$.

	Glu			Meth		
	EXP	TP	ETP	EXP	TP	ETP
M-DS$_1$	52.37	26.29	63.40	79.61	56.52	49.50
M-DS$_1$-C		38.65			57.57	
M-DS$_2$	30.92	22.30	35.07	71.97	47.94	52.41
M-DS$_2$-C		30.92			43.18	
M-DS$_3$	37.68	21.59	33.95	94.22	48.50	75.11
M-DS$_3$-C		30.92			71.97	
M-DS$_4$	31.58	30.73	50.48	115.01	61.03	84.51
M-DS$_4$-C		23.14			86.36	
$\otimes$		0.35	**0.17**		0.40	**0.27**

Table 4: (Table 3 continued) Concentrations of metabolites in the synthetic wet-lab study. The weighted sample concentration is given in the -C rows each. The estimate of the expert EXP, TP and ETP are given in the columns. All concentrations are given in μ mol. Considering the median relative error ($\otimes$) of the concentration estimates the ETP approach is best in all cases. In a case by case comparison ETP is almost always the best, with three exceptions $\{(Spec_2, Val), (Spec_4, Glu), (Spec_1, Ile)\}$.

7.3 Non negative least squares fitting

The targets $f'_j(\bowtie)$ are now given in the functional description of (2) with optimized Δ_i, using the known Θ_k and our functional shape estimation for all peak groups. The function to fit is our reduced spectrum $s'(\bowtie)$. We add constraints for non negative α_i and allow for user definition of α_j fixed on a target f_j by employing standard optimization modeling techniques. Solving the optimization problem by use of a standard constrained linear least squares algorithm we obtain the α_j in a column vector α, which can subsequently be used to calculate the concentration estimates. To this end, the area under the α-scaled target is calculated and associated to the area of the α-scaled reference signal (here DSS). A scaling step is then performed, based on the number of protons ^{1}H present in the reference, nine for

[2]If we assume a spectral resolution of $SR = 0.5$ Hz, a device frequency of $F = 700$ MHz and an error of PPM-E= ± 0.01 ppm the number of valid positions V is $V \approx 2 \cdot \text{PPM-E}/(SR/F)$.

DSS, compared to the number of protons present in the metabolite e.g. four for Ala. This leads to the following equation for the concentration c in mol: $c(Ala) = \frac{area(Ala) \cdot c(DSS) \cdot 9}{area(DSS) \cdot 4}$ with *area* as an appropriate estimation function for the area under the curve. One can also calculate estimates of the lower concentration limits by scaling the target intensities of f'_j to the noise level and repeating the procedure. The reconstruction s^* is obtained as:

$$s^* = R^\top \cdot \alpha \tag{5}$$

To judge the fitness of this solution we may now either use the quality of fit provided by the LSQO algorithm or evaluate the reconstructed spectrum $s^*(\nu)$ with respect to $s(\nu)$ using a problem specific distance measure. Here we use either the standard Euclidean distance (EUC) or a functional distance measure as an extension of the L^p norm proposed in ([12]) (FUNC). The functional distance measure has the advantage of taking the functional nature of the spectra into account. The standard Euclidean distance considers the individual features of the NMR spectrum to be independent, so that a change in the order of the ppm positions does not affect the calculated distance. However, the features or measurement points in NMR spectra are not independent, so that a distance taking this aspect into account can be considered to be more appropriate for this type of data. Lee proposed a distance measure taking the functional structure into account by involving the previous and next values of a signal v_i in the i-th term of the sum, instead of v_i alone. Assuming a constant sampling period τ, the proposed norm (FUNC) is:

$$\mathcal{L}_p^{fc}(\mathbf{v}) = \left(\sum_{k=1}^{D} \left(A_k(\mathbf{v}) + B_k(\mathbf{v}) \right)^p \right)^{\frac{1}{p}} \tag{6}$$

with

$$A_k(\mathbf{v}) = \begin{cases} \frac{\tau}{2}|v_k| & \text{if } 0 \leq v_k v_{k-1} \\ \frac{\tau}{2} \frac{v_k^2}{|v_k| + |v_{k-1}|} & \text{if } 0 > v_k v_{k-1} \end{cases} \tag{7}$$

$$B_k(\mathbf{v}) = \begin{cases} \frac{\tau}{2}|v_k| & \text{if } 0 \leq v_k v_{k+1} \\ \frac{\tau}{2} \frac{v_k^2}{|v_k| + |v_{k+1}|} & \text{if } 0 > v_k v_{k+1} \end{cases} \tag{8}$$

representing the triangles on the left and right sides of v_i and D being the data dimensionality. For the data considered in this paper v takes the position of ν. As for L_p, the value of p is assumed to be a positive integer. At the left and right extremes of the sequence, v_0 and v_D are assumed to be equal to zero. The concept of the L^p-norm is shown in Figure 5. The calculation of this norm is slightly more complex than that of the standard Euclidean but, as is shown below, significantly improves the fitting results as well as the convergence speed of the GA[3].

8 Results and Discussion

8.1 Identification and quantification

We have tested our approach using measurements of metabolites in lysates of cultured cells as well as a small test set with known concentrations of defined metabolites. Rather than focusing on a specific biochemical question we aim to compare the range and concentrations of metabolites detected using TP and ETP with those obtained by manual expert profiling.

[3]The additional effort in the calculations is almost negligible - the time to calculate a generation is changing only minor, by a few seconds.

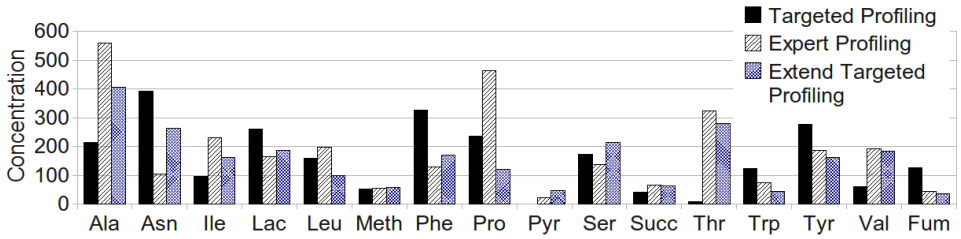


Figure 6: Concentration estimates for some of the different metabolites using ETP in comparison to TP and an expert analysis (Spec_1). The x-axis denotes the metabolites and the y-axis the intensities in μ-mol.

8.1.1 Wet lab metabolite mixture experiment

The wet-lab mixture data sets (M-DS) can be considered to be an artificial data set with known concentrations. In Table 3 the known concentrations (weighted sample) and the concentration estimates as obtained by the expert, TP and ETP using the functional distance measure are given. We observe that the ETP approach is closer to the expert estimation than is TP. The DSS concentration was given with 77.05 μ mol for all spectra.

8.1.2 Cell culture experiment

Details of the analyzed cell extract data are shown in Figure 6. The optimized approach provides results which are much closer to the expert analysis, for 16 of the 21 targets. Eth, Cit, His, Myo and Mal were noted as being absent by the expert but by ETP and TP with very low concentrations.

Test spectrum	Error TP	Error ETP (FUNC)	Error ETP (EUC)
Spec_1	49.65	31.95 (97)	32.05 (121)
Spec_2	68.68	45.89 (106)	68.71 (122)
Spec_3	30.92	28.40 (112)	29.87 (147)
Spec_4	87.53	55.70 (118)	56.01 (132)
Spec_5	64.04	47.30 (97)	46.39 (121)
Spec_6	111.09	87.60 (81)	93.77 (137).

Table 5: Mean errors in μ-mol of TP and ETP with respect to the expert concentration estimates. The expert concentration is assumed to be optimal (0 error), the values for TP and ETP are then compared with the expert using the mean square error, normalized by the number of metabolites. It can be seen that the new approach clearly improves the concentration estimates. The number of generations until convergence is shown in brackets.

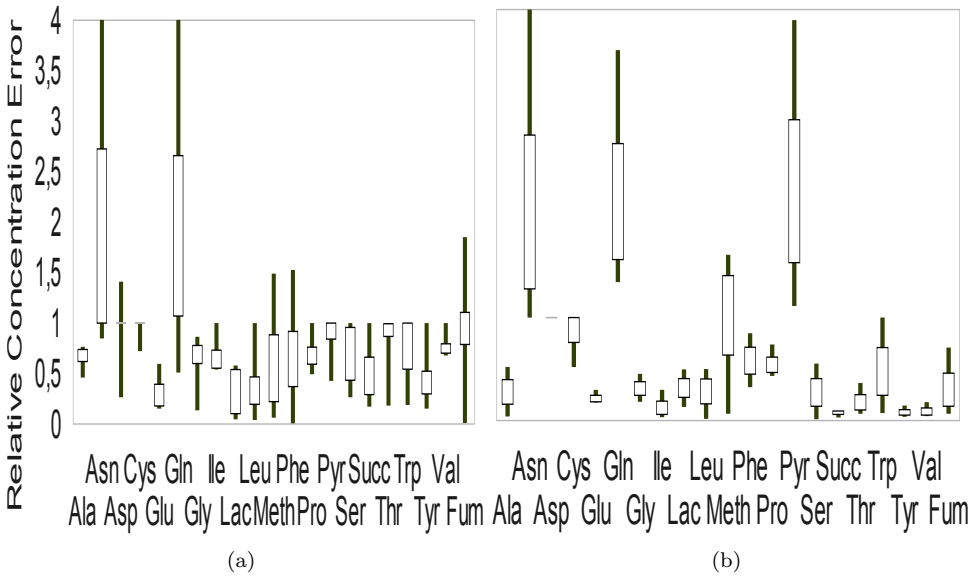
Figure 9 shows a reconstruction of a signal part with respect to the original signal to illustrate the effect of the shift correction.

From Table 5 we observe that the EUC measure in the fitness function is indeed less effective then the FUNC measure, consistent with our expectation that the FUNC norm is more appropriate to data which are themselves functions [4]. We subsequently restrict

[4]Using the median in Figure 6 gives similar results with respect to TP but comparing FUNC and EUC the results are less pronounced due to the dominance of the (many) small metabolites

	Ala	Asn	Asp	Cys	Glu	Gln	Gly	Ile
TP	0.66	1.77	1	1	0.26	1.83	0.69	0.61
ETP	0.28	1.56	1	1	0.19	1.78	0.3	0.08
Imp	1	1	0	0	1	1	1	1
Sig	+	o	o	o	o	o	+	+
	Lac	Leu	Meth	Phe	Pro	Pyr	Ser	Succ
TP	0.34	0.27	0.49	0.56	0.64	1	0.64	0.44
ETP	0.32	0.31	1.21	0.58	0.5	1.96	0.26	0.09
Imp	1	1	2	2	1	2	1	1
Sig	o	o	o	o	o	o	o	o
	Thr	Trp	Tyr	Val	Fum	Mean		
TP	0.97	0.85	0.44	0.71	1	0.81		
ETP	0.14	0.42	0.07	0.06	0.21	0.66		
Imp	1	1	1	1	1	-		
Sig	+	o	o	+	o	0		

Table 6: Relative median metabolite error. Change judged in the row labeled by *Imp*: 1 (improvement/optimal), 2 (worse estimate), 0 (no improvement). The rows labeled with *Sig* indicate if the change was significant by use of a $t-$test.



(a) (b)

Figure 7: Relative error estimates for different metabolites using TP (a) and ETP (b). The y-axis encodes the relative error and is limited to [0,4]. The x-axis lists the different metabolites.

our analysis to the FUNC norm and the standard TP approach. In Figure 7 we show Box-Whisker Plots of the relative concentration errors with respect to the expert of the metabolite concentrations using the standard TP and the ETP (FUNC) approaches. It can be seen that the relative error of ETP is much smaller than that of TP in the large majority of the metabolites. Also the variance of the results is smaller. Median errors of TP vs ETP are shown in Table 6. Significance of an improvement is indicated by a + using a t-test on a 5% level, with significances of $p \leq 0.03$. We observe that $\approx 25\%$ of the differences are significant and all of these are positive.

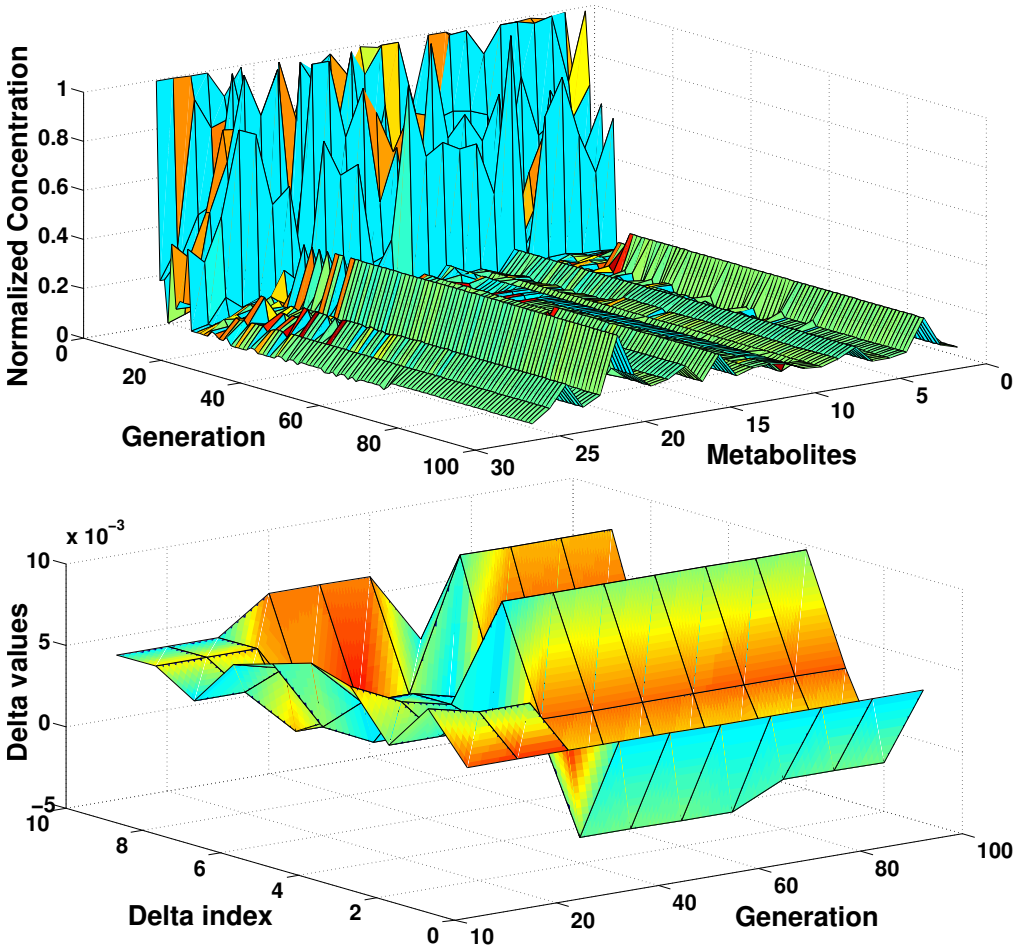


Figure 8: (top): Concentrations (normalized) for metabolites over time (without CSI) for Spec_1. (bottom): parameter changes over time for $\Delta_1 \ldots \Delta_{10}$ using the median over 10 generations.

8.2 Shift uncertainty properties and influence

The shift uncertainty estimates Δ change over time with respect to the GA evolution and the underlying constraints. The GA can only determine a local optimal solution, which is expected to correspond to the global optimum in only a very few cases. An analysis of the number of updates per shift uncertainty estimate reveals parameters which are likely to be incorrect either because they have not been updated at all or because they have been updated very frequently. An example for Spec_1 is shown in Figure 11. Taking this statistic into account the expert can be assisted by an indicator that highlights peak group shifts, that are likely to be incorrect. Considering the values of the shift uncertainties for the analyzed spectra we found around $3 - 4\%$ of the Δ_i to be 0 after convergence. Analyzing the shift updates also provides information about potentially unreliable modeled regions, indicated by either very few or very many Δ updates, as shown in Figure 10. There the relative number of Δ changes by the GA with respect to the total number of generations until convergence is shown over all spectra. The various metabolites are indicated by different symbol shapes

and shadings.

One can clearly see that for most Δ (indicated by the symbols) around 40% of the GA generations are sufficient to obtain a stable solution. Even if this solution may not be a global optimum, it can still be considered as a stable local optimum. For some of the Δ (e.g. those for pyruvate and aspartate) a (much) larger number of updates is necessary and it can be expected that these shifts are not well optimized, but that no better solution could be found by the GA. For some of the other metabolites one can also see that only a single group is optimized very frequently as is the case for the group of Val (valine) around 0.98 ppm or Ser (serine) around 3.83 ppm. The concentration estimates for these two amino acids compare quite well with the expert estimates. Very few updates can be observed e.g. for Succ (succinate) around 2.39 ppm, Gly (glycine) around 3.55 ppm or Glu (glutamate) around 2ppm and 3.75ppm. Interestingly Succ, Gly and Glu are optimized very well and the estimated concentrations correspond reasonably to those obtained by the expert. However it should be borne in mind that the concentration estimate is not equally split over the groups.

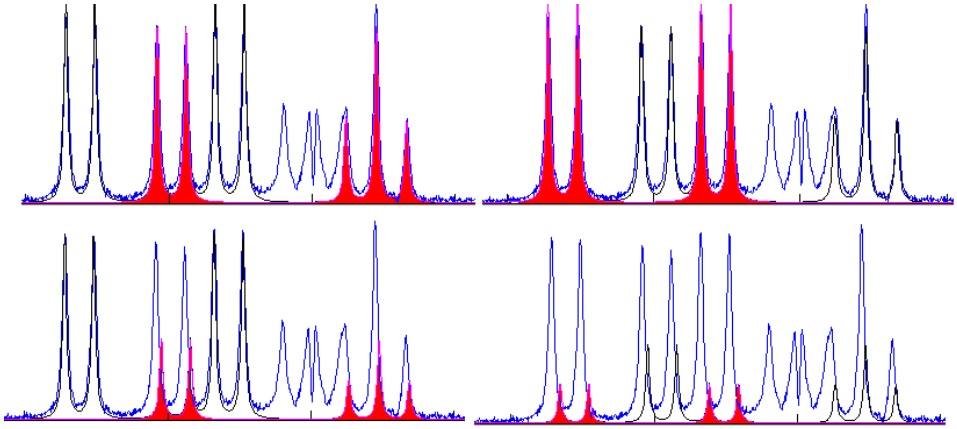


Figure 9: Spectrum in the region of valine and iso-leucine. The two sub figures on the top show the fit with ETP, left for iso-leucine (filled), right for valine (filled). Below the same but in the original TP fit.

The plot in Figure 10 provides an initial indication of which metabolites are most likely to have been poorly optimized and should therefore be manually corrected by the expert. This provides a basis for the focused and guided manual interaction avoiding the inspection of all metabolites. In the example shown, the optimizations appear to have been reasonably effective and correct for those metabolites for which the number of updates for the corresponding Δ lies within a range of $20 - 40\%$. The plots in Figure 8 show the effect of the genetic evolution with respect to the concentration of the metabolites and the parameter modifications. One can see that most of the optimization of GA parameters and hence of concentration changes occurs in the first 10-20 generations. The plot also shows that even relatively small errors in the Δ_i may have large impact on the concentration estimates, with very high values at the beginning of the optimization and comparatively small values at the end for some metabolites.

Median relative error estimates of single metabolites using TP and ETP (FUNC) are shown in the Box-Whisker plot 7. The relative error is calculated as the absolute concentration error compared to the expert value.In terms of the median errors, we find that ETP provides a clear improvement over TP but has still problems with some metabolites such as

Leu, Meth, Phe and Pyr. In these cases, however, we note that even the manual fit by the expert is challenging. On average the median error improved from 0.78 to 0.64 with 0 as the perfect agreement. These findings show that ETP is superior to TP in providing reasonable estimates for metabolites on a magnitude level. However, the accuracy attainable from single measurements is still low. This highlights the need both for the use of experimental replicates and for the analysis of multiple spectra of the same sample.

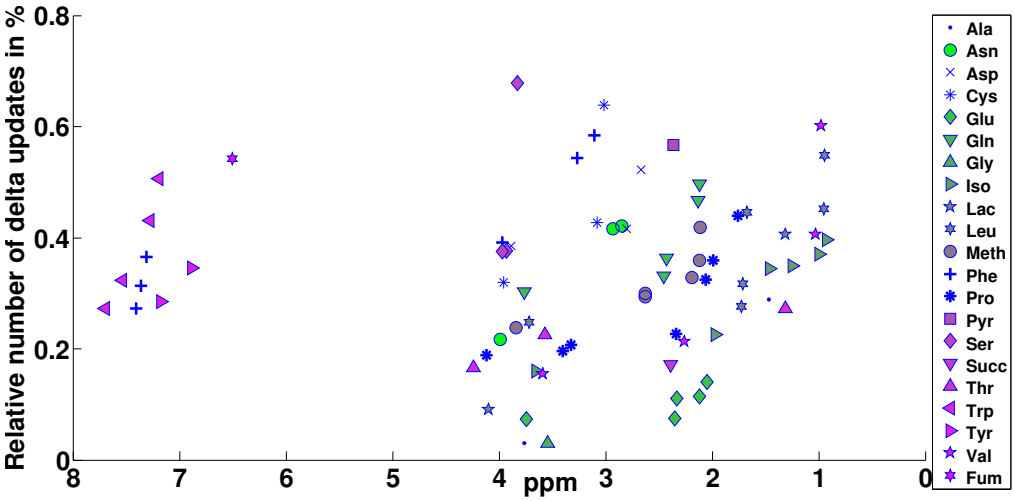


Figure 10: Relative #Δ updates in % (y) with respect to the ppm position (x).

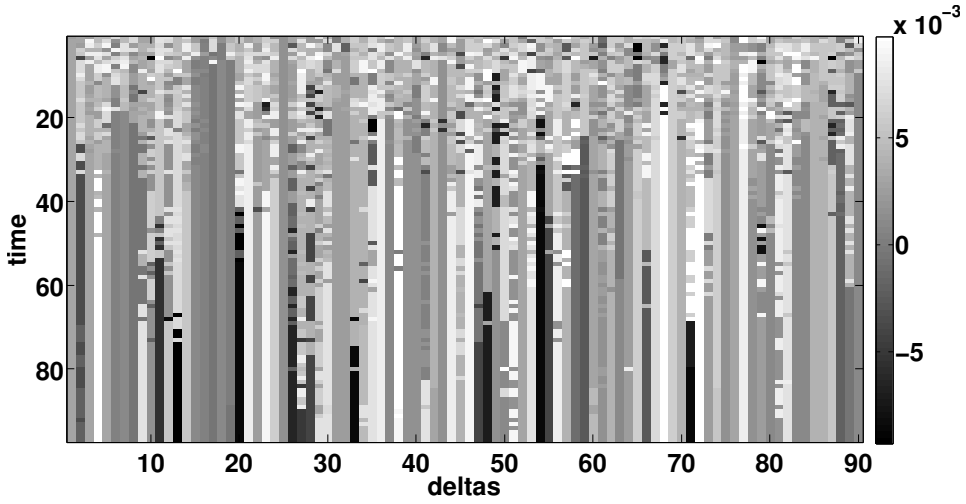


Figure 11: A typical evolution of the Δ (columns) over time (rows) for 97 generations. The gray levels indicate the shift values. Some Δ converge early to stable values, some (few) need more updates.

9 Conclusion

In summary, this work has shown that an approach combining GAs with LSQO leads to highly effective error estimates for the shift uncertainties in ^{1}H NMR measurements. The simultaneous fit outperforms the standard TP approach with respect to identification and quantification accuracy and compares favorably to the expert analysis. We have further shown that the usage of a data specific (functional) distance measure to calculate the fitness values is preferable to a standard Euclidean measure. It also significantly improved the convergence rate of the GA. The interpretation of the obtained shifts over time with the best model allows an in depth analysis of the optimization, revealing potentially unreliable fits. This provides initial guidance for the expert to focus further manual improvement of the obtained fit where necessary, reducing the demand for extensive shift corrections in order to generate correct uncertainty estimates. Furthermore, the approach also allows the manual, specification of concentration values in the fit for known concentrations, by additional constraints. Overall the combined approach can improve the identification and quantification accuracy of NMR based targeted profiling to allow a semi-automatic high throughput analysis. Further improvements are to be expected from improved preprocessing of the spectra. Variations in the baseline and slightly incorrect lineshapes being the main sources of error in the automatic identification and quantification of metabolites in NMR measurements.

Acknowledgment

We thank Prof. Thomas Villmann (Univ. of Appl. Sc. Mittweida) for discussions about functional signal processing, the METASTEM team and Peter Tino, Univ. of Birmingham for a very effective research stay during the preparation of this manuscript.

Funding: This work was supported by the Fed. Ministry of Edu. and Res.:FZ:0313833 A, (NMR Metabolic Profiling of the Stem Cell Niche, METASTEM), the German Res. Fund. (DFG), HA2719/4-1 (Relevance Learning for Temporal Neural Maps) and by the Cluster of Excellence 277 Cognitive Interaction Technology funded in the framework of the German Excellence Initiative.

References

[1] S. Böcker, Matthias C. Letze, Zsuzsanna Liptak, and Anton Pervukhin. Sirius: decomposing isotope patterns for metabolite identification. *Bioinformatics*, 25(2):218–224, 2009.

[2] G. Brelstaff, Manuele Bicego, Nicola Culeddu, and Matilde Chessa. Bag of peaks: interpretation of nmr spectrometry. *Bioinformatics*, 25(2):258–264, 2009.

[3] D. Chang, C. D. Banack, and S. L. Shah. Robust baseline correction algorithm for signal dense nmr spectra. *Journal of Magnetic Resonance*, 187:288–292, 2007.

[4] M. Defernez and I. J. Colquhoun. Factors affecting the robustness of metabolite fingerprinting using 1H-NMR spectra. *Phytochemistry*, 62:1009–1017, 2003.

[5] R. Fletcher. *Practical Methods of Optimization*. Wiley VCH, 2000.

[6] J. Forshed, I. Schuppe-Koistinen, and S. P. Jacobsson. Peak alignment of NMR signals by means of a genetic algorithm. *Analytica Chimica Acta*, 487:189–199, 2003.

[7] J. Forshed, R. J. O. Torgrip, K. Magnus Aberg, B. Karlberg, J. Lindberg, and S. P. Jacobsson. A comparison of methods for alignment of NMR peaks in the context of cluster analysis. *Journal of Pharmaceutical and Biomedical Analysis*, 38:824–832, 2005.

[8] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison Wesley, Reading, MA, 1989.

[9] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer, New York, 2001.

[10] R. M. Jarvis and R. Goodacre. Genetic algorithm opimization for pre-processing and variable selection of spectroscopic data. *Bioinformatics*, 21:860–868, 2005.

[11] H.-W. Koh, J. Lambert, S. Maddula, R. Hergenrder, and L. Hildbrand. Feature selection by lorentzian peak reconstruction for 1-h nmr post processing. In *Proc. of CBMS 2008*, pages 608–613. IEEE Press, 2008.

[12] J. Lee and M. Verleysen. Generalizations of the lp norm for time series and its application to self-organizing maps. In Marie Cottrell, editor, *5th Workshop on Self-Organizing Maps*, volume 1, pages 733–740, 2005.

[13] M. H. Levitt. *Spin Dynamics: Basics o Nuclear Magnetic Resonance (2nd Ed.).* Wiley, 2008.

[14] Mathworks Inc. Matlab 2008b, 2008.

[15] Pedro Mendes. Metabolomics and the challenges ahead. *Briefings in Bioinformatics*, 7(2):172, 2006.

[16] M. Mitchell. *An Introduction to Genetic Algorithms.* MIT Press, Boston, MA, 1995.

[17] S. Moco, R. J. Bino, Ric C.H. De Vos, and J. Vervoort. Metabolomics technologies and metabolite identification. *Trends in Analytical Chemistry*, 26(9):855–866, 2007.

[18] K. M. Pierce, B. W. Wright, and R. E. Synovec. Unsupervised parameter optimization for automated retention time alignment of severely shifted gas chromatography data using the piecewise alignment algorithm. *Journal of Chromatography A*, 1141:106–116, 2007.

[19] F. Savorani, G. Tomasi, and S.B. Engelsen. icoshift: A versatile tool for the rapid alignment of 1D NMR spectra. *Journal of Magnetic Resonance*, 202:190–202, 2010.

[20] F.-M. Schleif. *Prototype based Machine Learning for Clinical Proteomics.* PhD thesis, Technical University Clausthal, Technical University Clausthal, Clausthal-Zellerfeld, Germany, 2006.

[21] F.-M. Schleif. Preprocessing of nuclear magnetic resonance spectrometry data. *Machine Learning Reports*, 1(MLR-01-2007), 2007. ISSN:1865-3960.

[22] F.-M. Schleif, T. Riemer, M. Cross, and T. Villmann. Automatic identification and quantification of metabolites in h-nmr measurements. In *In Proceedings of the Workshop on Computational Systems Biology (WCSB) 2008*, pages 165–168, 2008.

[23] T. Schlumm and T. Riemer. Nmrj: A feasability study for a fully Java based platform independent MR processing and analysing program. In *Proceedings of the International Society for Magnetic Resonance in Medicine*, volume 9, page 798, Glasgow, Mai 2001.

[24] S.A. Smith, T.O. Levante, B.H. Meier, and R.R. Ernst. Computer simulations in magnetic resonance. an object oriented programming approach. *J. Magn. Reson.*, 106a:75–105, 1994.

[25] H. Son, G. Hwang, K. Kim, H. Ahn, W. Park, F. Berg, Y. Hong, and C. Lee. Metabolomic studies on geographical grapes and their wines using 1H NMR analysis coupled with multivariate statistics. *Journal of Agric. Food Chem*, 57(4):1481–1490, 2009.

[26] J. Swire, S. Fuchs, J. Bundy, and A. Leroi. The cellular geometry of growth drives the amino acid economy of caenorhabditis elegans. *Proceeding of the Royal Society B*, 276(1668):2747–2754, 2009.

[27] S. Tiziani, A. Lodi, F. Khanim, M. Viant, C. Bunce, and U. Günther. Analysis of mixed lipid extracts using 1h nmr spectra. *PLoS ONE*, 4(1), 2009.

[28] J. T. W. E. Vogels, A.C. Tas, J. Venekamp, and J. v. d. Greef. Partial linear fit: a new NMR spectroscopy preporcessing tool for pattern recognition applications. *Journal of Chemometrics*, 10:425–438, 1996.

[29] W. Weckwerth. Metabolomics in systems biology. *Ann. Rev. Plant Biol.*, 54:669–689, 2003.

[30] A. M. Weljie, J. Newton, P. Mercier, E. Carlson, and C. M. Slupsky. Targeted profiling: Quantitative analysis of 1h nmr metabolomics data. *Analytical Chemistry*, 78:4430–4442, 2006.

[31] Y. Xi and D. M Rocke. Baseline correction for nmr spectroscopic metabolomics data analysis. *BMC Bioinformatics*, 9:324–333, 2008.

[32] J. Xia, T. C. Bjorndahl abd P. Tang, and David S Wishart. Metabominer semi-automated identification of metabolites from 2d nmr spectra of complex biofluids. *BMC Bioinformatics*, 9:507–522, 2008.

[33] Q. Zhao, R. Stoyanova, S. Du, P. Sajda, and T. R. Brown. Hiresa tool for comprehensive assessment and interpretation of metabolomic data. *Bioinformatics*, 22(20):2562–2564, 2006.

Chapter 4

Large scale models

4.1 Efficient kernelized prototype based classification

The article *Efficient kernelized prototype based classification* by F.-M.**Schleif**, T. Villmann, B. Hammer and P. Schneider was published by Neural Systems 21 (6), p. 443 - 457, in 2011 In the article an extension of Kernelized Generalized Learning Vector Quantization (KGLVQ) is proposed employing a sparsity and approximation technique to reduce the learning complexity. Generalization error bounds and experimental results on different real world image processing tasks are shown. I derived the theoretical extensions and implemented the algorithms. B. Hammer supported the error bound analysis. P. Schneider and T. Villmann provided relevant suggestions in early discussions about this work and the initial conference contribution. I wrote the article. All authors discussed the general article.

Additional publications in international conferences where I am co-author and which cover a similar or related topic include:

1. F.-M. **Schleif**, A. Gisbrecht and B. Hammer, *Accelerating Kernel Neural Gas*, Proceeding of ICANN'2011, 150–158, 2011 (Content: The kernelized Neural Gas is extended by the Nyström approximation to obtain memory and runtime efficient behavior for large scale problems)

2. A. Gisbrecht, F.-M. **Schleif**, X. Zhu and B. Hammer, *Linear Time Heuristics for Topographic Mapping of Dissimilarity Data*, In Proceedings of Intelligent Data Engineering and Automated Learning (IDEAL)'2011, 25-33, 2011 (Content: Relational topographic mapping is extended by different approximation techniques to improve efficiency for large scale problems.)

3. A. Gisbrecht, B. Hammer, F.-M. **Schleif** and X. Zhu, *Accelerating kernel clustering for biomedical data analysis*, In Proceedings of IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)'2011, 154-161, 2011 (Content: Some clustering algorithms for dissimilarity learning have been extended by approximation strategies)

Efficient kernelized prototype based classification

F.-M. Schleif[1]
Dept. of Techn., Univ. of Bielefeld, Universitätsstrasse 21-23
33615 Bielefeld, Germany
E-mail: schleif@informatik.uni-leipzig.de

Thomas Villmann
Faculty of Math./Natural and CS, Univ. of Appl. Sc. Mittweida, Technikumplatz 17
09648 Mittweida, Germany
E-mail: villmann@hsmw.de

Barbara Hammer
Dept. of Techn., Univ. of Bielefeld, Universitätsstrasse 21-23
33615 Bielefeld, Germany
E-mail: bhammer@techfak.uni-bielefeld.de

Petra Schneider
University of Birmingham, School of Clinical & Experimental Medicine
Birmingham B15 2TT, United Kingdom
E-mail: p.schneider@bham.ac.uk

Abstract

Prototype based classifiers are effective algorithms in modeling classification problems and have been applied in multiple domains. While many supervised learning algorithms have been successfully extended to kernels to improve the discrimination power by means of the kernel concept, prototype based classifiers are typically still used with Euclidean distance measures. Kernelized variants of prototype based classifiers are currently too complex to be applied for larger data sets. Here we propose an extension of Kernelized Generalized Learning Vector Quantization (KGLVQ) employing a sparsity and approximation technique to reduce the learning complexity. We provide generalization error bounds and experimental results on real world data, showing that the extended approach is comparable to SVM on different public data.

1 INTRODUCTION

The dramatic growth in data generating applications and measurement techniques has created many high-volume and high-dimensional data sets. Most of them are stored

[1]corresponding author

digitally and need to be efficiently analyzed to be of use. Clustering and classification methods are very important in this setting and have been extensively studied in the last decades [17, 35, 33, 1, 24, 36, 10]. Challenges are mainly in the timely, memory efficient and accurate processing of such data also in the case of non linearly separable data with multiple thousand items.

Kernelized learning vector quantization (KGLVQ) was proposed in the approach[28] as an extended approach of Generalized Learning Vector Quantization (GLVQ) [29] with the goal to provide modeling capabilities for learning vector quantizers and to improve the performance in classification tasks. While the approach was quite promising it has been used only rarely due to its complexity. One challenge is the storage of a large kernel matrix and additionally the storage and update of a combinatorial coefficient matrix Ψ, implicitly representing the prototypes. This makes the approach inapplicable already for data with a comparably small number of items.

Data analysis using kernel methods is an active field of research [3, 6, 25] offering solutions for the analysis of complex problems. The involved kernel matrix needs, in its *original* form, quadratic space with the number of samples and involves usually cubic time complexity which can be quite demanding for large problems. This pose a big challenge on practical applications.

Modern approaches in discriminative kernel based learning like Sequential Minimal Optimization[27] and other try to avoid the direct storage and usage of the full kernel matrix or restrict the underlying optimization problem to subsets thereof[27, 35, 37]. For the KGLVQ approach such a strategy has not been proposed so far.

The Nyström-Approximation of Gram matrices constitutes a classical approximation scheme [41, 22], permitting the estimation of the kernel matrix by means of a low dimensional approximation. We will employ this method for the approximation of the distance calculations based on the kernel matrix as the key element of our accelerated kernelized GLVQ (AKGLVQ). A further issue with kernel methods is the model complexity by means of the stored data points. In case of the well known support vector machine (SVM) [38], these are the so called support vectors (SV). The number of SVs can become quite large for complex problems. Novel learning methods for the SVM try to shrink this value [19].

In case of KGLVQ the prototypes are implicitly modeled by a coefficient matrix over all data points which is typically dense. This however is not necessary for most data sets.

Sparsity is a natural concept in the encoding of data [26] and can be used to obtain compact sparse models. This concept has been used in many machine learning methods [21, 16] and different measures of sparsity have been proposed [26, 16]. Taking this into account we propose to integrate a sparsity constraint into KGLVQ allowing the explicit control of the sparsity of the coefficient matrix.

Both optimization concepts, Nyström and sparsity, are used to improve the complexity of KGLVQ such that it becomes applicable for large data sets.

In Sec. 2 we present a short introduction into kernels and give the notations used throughout the paper. Subsequently we present the KGLVQ algorithm and its approximated variant AKGLVQ by means of the Nyström approximation and the additional sparsity constraint. We show the efficiency of the novel approach for experiments on artificial and real life data. Finally, we conclude with a discussion.

2 PRELIMINARIES

We consider a set of vectors $\mathbf{v}_i \in \mathbb{X}^{\mathbb{D}}$ with $\mathbb{X}^{\mathbb{D}} \subseteq \mathbb{R}^{\mathbb{D}}$, $\mathbb{D}$ denoting the dimensionality and $|\mathbb{X}| = N$ the number of samples. Further we introduce prototypes $\mathbf{w}_j \in \mathbb{W}^{\mathbb{D}}$, with $|\mathbb{W}^{\mathbb{D}}| = M$ which induce a clustering of $\mathbb{X}^{\mathbb{D}}$ by means of their receptive fields consisting of the points $\mathbf{v}$ for which $d(\mathbf{v}, \mathbf{w_j}) \leq d(\mathbf{v}, \mathbf{w_l})$ holds for all $j \neq l$ and d denoting a

distance measure, typically the Euclidean distance. Further we introduce $c(\mathbf{v}) \in \mathcal{L}$ as the label of input $\mathbf{v}$, and $c(\mathbf{w})$ as the label of the prototype $\mathbf{w}$, respectively. $\mathcal{L}$ denotes the set of labels (classes) with $\#\mathcal{L} = N_{\mathcal{L}}$. Let $\mathbb{W}_c = \{\mathbf{w_l} | c(\mathbf{w}_l) = c\}$ be the subset of prototypes assigned to class $c \in \mathcal{L}$.

We also introduce two special notations for the prototype which is closest to a given point $\mathbf{v}_i$ with the same label: $\mathbf{w}^+$ or a different label: $\mathbf{w}^-$. The corresponding distance d_i^+, d_i^-:

$$
\begin{aligned}
d_i^+ &= d(\mathbf{w}^+, \mathbf{v}_i) \text{ with } \mathbf{w}^+ \in \mathbb{W}_c,\ c = c(\mathbf{v}_i), && (1)\\
\mathbf{w}^+ &:= \mathbf{w}_l : d(v_i, w_l) \leq d(v_i, w_j), \{\mathbf{w}_j, \mathbf{w}_l\} \in \mathbb{W}_c && (2)\\
d_i^- &= d(\mathbf{w}^-, \mathbf{v}_i) \text{ with } \mathbf{w}^- \notin \mathbb{W}_c,\ c = c(\mathbf{v}_i) && (3)\\
\mathbf{w}^- &:= \mathbf{w}_l : d(v_i, w_l) \leq d(v_i, w_j), \{\mathbf{w}_j, \mathbf{w}_l\} \notin \mathbb{W}_c
\end{aligned}
$$

Equation (2) is sometimes also referred as the *winner takes all* (wta) rule restricted to the $\mathbf{w}$ of the same class as $\mathbf{v}$.

Complex data are often not linearly separable in the Euclidean space and it was suggested to map the data $\mathbb{X}$ into a high dimensional Hilbert space $\mathbb{H}$ using a mapping function $\phi : \mathbb{X} \to \mathbb{H}$ to separate the data in a linear manner [33]. The explicit definition of an appropriate mapping ϕ can be complex for the high-dimensional feature space. As pointed out in [33] this explicit formulation is often not necessary, if we are able to express the calculation in our learning algorithm by means of inner products. If we have a positive semi-definite inner product function $\kappa(\mathbf{v}, \mathbf{v}')$, fulfilling the Mercer conditions we can expand it by means of its eigenvalues and eigenfunctions:

$$
\kappa\left(\mathbf{v}, \mathbf{v}'\right) = \sum_{i}^{\infty} \lambda_i \phi_i(\mathbf{v}) \phi_i(\mathbf{v}') = \left\langle \phi\left(\mathbf{v}\right), \phi\left(\mathbf{v}'\right) \right\rangle_{\mathcal{F}} \tag{4}
$$

Now we can express the inner products in the feature space based on the kernel function κ calculated in the Euclidean space using e.g. a Gaussian kernel $\kappa(\mathbf{v}, \mathbf{v}') = \exp(-\|\mathbf{v} - \mathbf{v}'\|^2/\sigma^2)$ [8]. The calculation in the GLVQ can be done based on inner products such that (4) is applicable as used to derive the KGLVQ[28].

3 ALGORITHM

Learning vector quantization (LVQ) is a supervised learning scheme. It was introduced as a generic concept for intuitive prototype-based classification algorithms [20]. Several variants were developed to improve the standard algorithms [13, 29, 34]. LVQ algorithms are based on the empirical risk minimization (ERM) principle and describe the data space by means of prototypical representants (vectors), which are in general elements of the original data space. The main benefit, beside of its good generalization performance [14], is the direct access to the model constituents by means of the prototypes. The prototypes can be directly inspected and provide human interpretable information about typical aspects of the represented data classes.

Generalized Learning Vector Quantization (GLVQ) is an extension of the standard LVQ providing a cost function [29] recently extended in two kernelized variants [28, 31]. It is a margin optimization method [7] and can inherently deal with multi class data. Moreover, it is effective also under different distance measures and objectives [11]. The kernelized variants of GLVQ, namely KGLVQ and differentiable kernelized GLVQ (D-KGLVQ) are effective extensions of the original GLVQ concept but suffer from its high complexity or limitations regarding the kernel choice [31]. Subsequently, we briefly review the concepts of GLVQ and KGLVQ which will be extended, by two optimization techniques, yielding AKGLVQ later on.

3.1 Standard GLVQ

The cost function for GLVQ is given as

$$E = Cost_{GLVQ} = \sum_{i}^{N} \mu(\mathbf{v_i}) \quad \mu(\mathbf{v_i}) = \frac{d_i^+ - d_i^-}{d_i^+ + d_i^-} \tag{5}$$

which is optimized with respect to the free parameters (here the prototypes), by stochastic gradient descent. Note that the classifier function $\mu(\mathbf{v})$ is positive if the vector $\mathbf{v}$ is misclassified and negative otherwise.

The learning rule of GLVQ is obtained taking the derivatives of the above cost function with respect to the parameters $\mathbf{w}$. Using $\frac{\partial \mu(\mathbf{v}_i)}{\partial \mathbf{w}^+} = \xi^+ \frac{\partial d_i^+}{\partial \mathbf{w}^+}$ and $\frac{\partial \mu(\mathbf{v}_i)}{\partial \mathbf{w}^-} = \xi^- \frac{\partial d_i^-}{\partial \mathbf{w}^-}$ with[2]

$$\xi^+ = \frac{2 \cdot d_i^-}{(d_i^+ + d_i^-)^2} \quad \xi^- = \frac{-2 \cdot d_i^+}{(d_i^+ + d_i^-)^2} \tag{6}$$

one obtains for the weight updates [12]:

$$\triangle \mathbf{w}^+ = \epsilon^+ \cdot \xi^+ \cdot \frac{\partial d_i^+}{\partial \mathbf{w}^+} \quad \triangle \mathbf{w}^- = \epsilon^- \cdot \xi^- \cdot \frac{\partial d_i^-}{\partial \mathbf{w}^-} \tag{7}$$

with $\epsilon^{+/-}$ as learning rates, which are typically in the range of 10^{-5}.

3.2 Kernelized GLVQ

We now briefly review the main concepts used in Kernelized GLVQ (KGLVQ) as given in the paper of Qin[28]. The KGLVQ makes use of the same cost function as GLVQ but with the distance calculations done in the kernel space. Under this setting the prototypes cannot explicitly be expressed as vectors in the feature space due to lack of knowledge about the feature space. Instead Qin[28] models the feature space as a linear combination of all images $\phi(\mathbf{v})$ of the datapoints $\mathbf{v}$. Thus a prototype vector may be described by some linear combination of the feature vectors: $\mathbf{w}_j = \sum_{l=1}^{N} \psi_{j,l} \phi(\mathbf{v}_l)$, $\psi_j \in \mathbb{R}^N$ is the corresponding coefficient vector. The distance in feature space for a given $\phi(\mathbf{v}_i)$ and $\mathbf{w}_j$ is computed as:

$$\begin{aligned} d_{i,j}^2 &= \|\phi(\mathbf{v}_i) - \mathbf{w}_j\|^2 = \|\phi(\mathbf{v}_i) - \sum_{l=1}^{N} \psi_{j,l} \phi(\mathbf{v}_l)\|^2 \\ &= k(\mathbf{v}_i, \mathbf{v}_i) - 2 \sum_{l=1}^{N} k(\mathbf{v}_i, \mathbf{v}_l) \cdot \psi_{j,l} \\ &\quad + \sum_{s,t=1}^{N} k(\mathbf{v}_s, \mathbf{v}_t) \cdot \psi_{j,s} \psi_{j,t} \end{aligned} \tag{8}$$

The update rules of GLVQ can be modified by substituting the Euclidean distance by Equation (8) and taking derivatives with respect to the coefficients $\psi_{j,l}$. The detailed equations are available in [28], a simplified version for the coefficient update is given later on. The final model consists of the pre-calculated kernel matrix and the combinatorial coefficient matrix for the ψ coefficients.

[2]Divisions including vectors are used element-wise throughout the paper.

3.3 Approximation of the kernel matrix by Nyström

As pointed out in the paper of Zhang[41], different strategies have been proposed to overcome the complexity problem caused by the kernel matrix K in modern machine learning algorithms. One promising approach is the Nyström approximation.

It originates from the numerical treatment of integral equations of the form $\int \mathcal{P}(y)k(x,y)\phi_i(y)dy = \lambda_i\phi_i(x)$ where $\mathcal{P}(\cdot)$ is the probability density function, k is a positive definite kernel function, and $\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$ are the eigenvalues with $\phi_1, \phi_2, \ldots$ the respective eigenfunctions of this integral equation. Given a set of i.i.d. samples $\{x_1, \ldots, x_q\}$ drawn from $\mathcal{P}(\cdot)$, the basic idea is to approximate the integral by the empirical average

$$1/q \sum_{j=1}^{q} k(x, x_j)\phi_i(x_j) \approx \lambda_i\phi_i(x)$$

which can be written as the eigenvalue decomposition: $K\phi = q\lambda\phi$. $K_{q\times q} = [K_{i,j}] = [k(x_i, x_j)]$ is the kernel matrix defined on X, and $\phi = [\phi_i(x_j)] \in \mathbb{R}^q$. Solving this equation we can calculate $\phi_i(x)$ as

$$\phi_i(x) \approx 1/(q\lambda) \sum_{j=1}^{q} k(x, x_j)\phi_i(x_j)$$

which is costly. To reduce the complexity, one may use only a subset of the samples which is commonly known as the Nystöm method.

Suppose the sample set $V = \{\mathbf{v}_i\}_{i=1}^N$, with the corresponding $N \times N$ kernel matrix K. We randomly choose a subset $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^q, \mathbf{Z} \subset V, q << N$ of landmark points and a corresponding kernel sub matrix $\mathbf{Q}_{q\times q} = [k(\mathbf{z}_i, \mathbf{z}_j)]_{i,j}$. We calculate the eigenvalue decomposition of this sub matrix: $\mathbf{Q}\phi_z = q\lambda_z\phi_z$ and obtain the corresponding eigenvector $\phi_z \in \mathbb{R}^q$ and the eigenvalue $q\lambda_z$. Subsequently we calculate the interpolation matrix $\hat{\mathbf{K}}_{N\times q} = [k(\mathbf{v}_i, \mathbf{z}_j)]_{i,j}$ to extend the result to the whole set V. We approximate the eigen-system of the full $K\phi_K = \phi_K\lambda_K$ by [39]:

$$\phi_K \approx \sqrt{\frac{q}{N}}\hat{\mathbf{K}}\phi_Z\lambda_{\mathbf{Z}}^{-1}, \lambda_K \approx \frac{N}{q}\lambda_{\mathbf{Z}}.$$

K can be subsequently reconstructed as

$$\begin{aligned} K &\approx \left(\sqrt{\frac{q}{N}}\hat{\mathbf{K}}\phi_Z\lambda_{\mathbf{Z}}^{-1}\right)\left(\frac{N}{q}\lambda_{\mathbf{Z}}\right)\left(\sqrt{\frac{q}{N}}\hat{\mathbf{K}}\phi_Z\lambda_{\mathbf{Z}}^{-1}\right)' \\ &= \hat{\mathbf{K}}\mathbf{Q}^{-1}\hat{\mathbf{K}}' \end{aligned}$$

To integrate the Nyström approximation into KGLVQ we only need to modify the distance calculation between a prototype $\mathbf{w_j}$ and a data point $\mathbf{v_i}$ which can be expressed using the Nyström approximation. In KGLVQ the prototypes are expressed by means of a linear combination of the datapoints in the feature space as shown in [28]. Hence it is sufficient to update the coefficients of this linear combination. The original update equation for the coefficient matrix in KGLVQ read as:

$$\psi_{\pm,\mathbf{r}'}^{t+1} = \begin{cases} [1 \mp \epsilon \cdot \frac{4\cdot d_i^{\mp}}{d_i^{\pm}+d_i^{\mp}}] \cdot \psi_{\pm,\mathbf{r}'}^t & \text{if } \mathbf{v}_{\mathbf{r}'} \neq \mathbf{v}_i \\ [1 \mp \epsilon \cdot \frac{4\cdot d_i^{\mp}}{d_i^{\pm}+d_i^{\mp}}] \cdot \psi_{\pm,\mathbf{r}'}^t & \\ +\epsilon \cdot \frac{4\cdot d_i^{\mp}}{d_i^{\pm}+d_i^{\mp}} & \text{if } \mathbf{v}_{\mathbf{r}'} = \mathbf{v}_i \end{cases}$$

with $t+1$ indicating the coefficient ψ after the update. A single prototype update has a complexity of $O(N^2)$, due to the double sum in (8). The index or superscript $\pm$ corresponds to the prototype with the same $(+)$ or different $(-)$ label as the data point

$\mathbf{v_i}$ as already defined previously. The point $\mathbf{v}_i$ is the current point used in the iterative gradient descend optimization. The index r' refers to the considered datapoint in the linear combination (the column index of Ψ). The KGLVQ update above is almost identical for the AKGLVQ but the distance calculations are done using the Nyström approximation with Equation (9):

$$d_{\cdot,i} = K(i,i) - 2 \cdot T_{\cdot,i} + \mathrm{diag}(\Psi \cdot T') \quad (9)$$

$$\text{with } T_{j,\cdot} = ((\psi_j \cdot \hat{\mathbf{K}}) \cdot \mathbf{Q}^{-1}) \cdot \hat{\mathbf{K}}' \quad (10)$$

where diag provides diagonal elements of the associated matrix. Using Nyström-approximation, the complexity in AKGLVQ is reduced to $O(q^3 + qN)$, caused by the SVD (for some recent work on SVD see[18]) to calculate the inverse of the matrix in the Nyström-approximation and the remaining distance calculation costs [39].

3.4 Sparse coefficient matrix

In the paper of Olshausen[26], sparsity has been found to be a natural concept in the visual cortex of mammals. This work motivated the integration of sparsity concepts into many machine learning methods to obtain sparse and efficient models. Here we will integrate sparsity as an additional constraint on the coefficient matrix Ψ such that the amount of non-zero coefficients is limited. This leads to a more compact descriptions of the prototypes, by means of a smaller linear mixture model. The used sparsity measure is the one as given in Olshausen[26]. The sparsity $\mathbb{S}$ of a row of α is measured as

$$\mathbb{S}(\psi_j) = -\sum_{l=1}^{N} S\left(\frac{\psi_{j,l}}{\sigma}\right) \quad (11)$$

with σ as a scaling constant. The function S can be of different type, here we use $S(x) = \log(1 + x^2)$. We extend the energy function of the KGLVQ by an additional term:

$$E_{AKGLVQ}(\gamma) = E_{KGLVQ}(\gamma) - \beta \mathbb{S}(\psi_{\mathbf{j}}) \quad (12)$$

The updates for the coefficients of $\mathbf{w}_i$ are structurally similar to those given in the standard KGLVQ using the Nyström formula to approximate the Gram matrix but include the additional term

$$\frac{\partial \mathbb{S}}{\partial \psi_{j,l}} = -\frac{2/\sigma^2 \cdot \psi_{j,l}}{1 + (\psi_{j,l}/\sigma)^2},$$

we restrict the coefficients to be $\psi_{j,i} \in [0,1]$ and bound them by $\sum_i \psi_{j,i} = 1$.

The effect of the sparsity constraint in AKGLVQ on the UCI iris data [4] with one prototype per class is shown in Figure 1. Both models achieve an accuracy of $\approx 90\%$ using a linear kernel. The sparsity constraint effectively helps to reduce the necessary memory of the matrix Ψ. Yet, the associated parameters have to be chosen adequately to balance sparsity and classification accuracy. The sparsity constraint could also be used to speed up the algorithm, by explicit omit operations involving multiplications with zero. This, however, requires a very careful and efficient implementation of the sparsity handling which is not easily accessible within the used runtime Matlab. During the classification step a sparse matrix Ψ can significantly limit the number of distance calculations necessary to map a new item in the feature space and to calculate the distance to a prototype. In the worst case with a dense matrix Ψ we get linear complexity $O(M \times N)$ to calculate the inner products for a new point, whereas a sparse matrix Ψ will typically scale in constant complexity $O(k \times M)$, assuming e.g. a k-approximation of the prototypes.

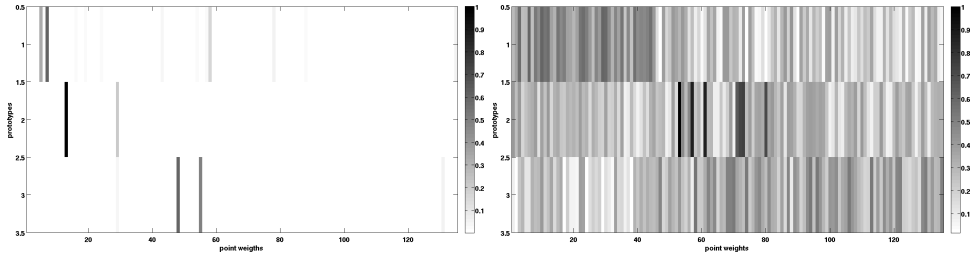


Figure 1: Effect of the sparsity constraint for the UCI iris data shown by means of the Ψ-matrix (normalized for better comparison). With sparsity (left), without sparsity right. Dark values indicated high loaded or high lighted data points for the considered prototype in the Ψ matrix. Data points with very low values over all prototypes can be safely removed from the model.

3.5 Generalization ability of KGLVQ

It has been shown in the approaches [7, 32] that generalization bounds for LVQ schemes can be derived based on the notion of the hypothesis margin of the classifier μ, independently of the input dimensionality. Rather the margin, i.e. the difference of the distance of points to its closest correct and wrong prototype, determine the generalization ability. This fact makes the algorithm particularly suitable for kernelization: essentially, the generalization ability transfers directly to the kernel version because of the fixed implicit embedding into the feature space. Thereby, large margin bounds are of particular interest due to the usually high dimensionality of the feature space. Bounds which depend on the number of free parameters would likely yield very weak bounds in such cases. For GLVQ as a large margin approach, a straightforward transfer of the bounds as provided in the approaches [7, 32] based on techniques as given in the article [2] is possible.

For convenience, we shortly review the setting as formalized e.g. in the derivation [32]. For simplicity, a classification by a kernelized prototype-based network into two classes is considered. We label prototypes corresponding to the two classes with $+$ and $-$, respectively. Classification takes place by a winner takes all rule (2), i.e., taking the kernel into account, a data point is mapped to the class

$$f : \mathbf{v} \mapsto \mathrm{sgn}\left(\min_{\mathbf{w}^+} \|\Phi(\mathbf{v}) - \mathbf{w}^+\| - \min_{\mathbf{w}^-} \|\Phi(\mathbf{v}) - \mathbf{w}^-\|\}\right) \tag{13}$$

where sgn selects the sign of the term. A trainable KGLVQ network corresponds to a function f in this class with M prototypes. We can assume that data $\mathbf{v}$ are bounded in size. Thus, also the images $\Phi(\mathbf{v})$ and the possible location of prototype vectors are bounded in size, we refer to the bound by B.

As usual, generalization bounds aim at limiting the generalization error $E_P(f) = P(f(\mathbf{v}) \neq c(\mathbf{v}))$ where P refers to a (probably unknown) probability distribution P. The margin of the classification is obtained by dropping the sign in (13) leading to the related function M_f. For a fixed positive value of the margin ρ and the associated loss

$$L : \mathbf{R} \to \mathbf{R}, t \mapsto \begin{cases} 1 & \text{if } t \leq 0 \\ 1 - t/\rho & \text{if } 0 < t \leq \rho \\ 0 & \text{otherwise} \end{cases}$$

a connection of the generalization error and the empirical error on m samples

$$\hat{E}_m^L(f) = \sum_{i=1}^{m} L(c_{\mathbf{v}} \cdot M_f(\mathbf{v}))/m \tag{14}$$

can be established with probability $\delta > 0$ simultaneously for all functions f using techniques of [2]:

$$E_P(f) \leq \hat{E}_m^L(f) + \frac{2}{\rho} R_m(M_{\mathcal{F}}) + \sqrt{\frac{\ln(4/\delta)}{2m}}$$

$R_m(M_{\mathcal{F}})$ denotes the so-called Rademacher complexity of the class of functions implemented by KLVQ networks with function M_f. The quantity can be upper bounded, using techniques of [32] and structural properties given in [2], by a term

$$\mathcal{O}\left(\frac{N^{2/3}B^3 + \sqrt{\ln(1/\delta)}}{\sqrt{m}}\right)$$

The quantity B depends on the kernel and can be estimated depending on the data distribution. Thus, generalization bounds for KGLVQ with arbitrary kernel result which are comparable to generalization bounds for GLVQ. Note that the only difference as compared to the derivation as provided in [32] consists in the fact that data are implicitly embedded in the feature space such that B depends on the given data points and the kernel.

4 EXPERIMENTS

We analyze our approach using artificial and real life data. The simulated data shall be considered as a toy data set to show the possibility to deal with non-linear separable data distributions, which is a typical application field of kernel methods. Subsequently we provide some analysis for very well known standard test data, followed by more complicated data sets which can not be processed by KGLVQ under reasonable time and memory settings. It should be pointed out that KGLVQ can be applied to very different types of problems, ranging from life-science data [40] to e.g. image processing tasks [23], as long as a valid kernel can be provided.

4.1 Simulated data

We start with the non-linear separable ring data set (DS1) and an RBF kernel in the distance measure. The data consist of 800 data points with 400 per ring in 2 dimensions as shown in Figure 2. The first ring has a radius of $r = 10$ and the second $r = 4$, points are randomly sampled in $[0, 2\pi]$. The data set has been normalized in $N(0, 1)$. We also analyzed the ring data using the additional sparsity constraint. In the original model 53% of the weights, averaged over the prototypes are almost 0 (values $\leq 1e-5$). In the sparsity approach we used σ^2 as the variance of the data scaled by 0.01 and a $\beta = 1$ and obtained a much sparser model with now 75% of the points close to zero.

4.2 Small sample size data

Now we present a comparison for 3 benchmark datasets taken from the UCI repository [4], namely the breast cancer data (wdbc), a diabetes study (pima) and the heart data set, used to predict a heart disease. All these data sets are two class examples with $N < 1000$, details are given in Table 1.

We analyze the performance of KGLVQ, AKGLVQ and SVM using the recently proposed Extreme Learning Kernel (ELM) [9]. The ELM kernel is actually a defacto parameter free kernel with the same classification performance as the RBF kernel with optimal σ [9], it has been fixed to $1e10$ in this study. SVM models are obtained by use of a Sequential Minimization Optimization (SMO) optimizer as proposed in [27] and the ELM kernel.

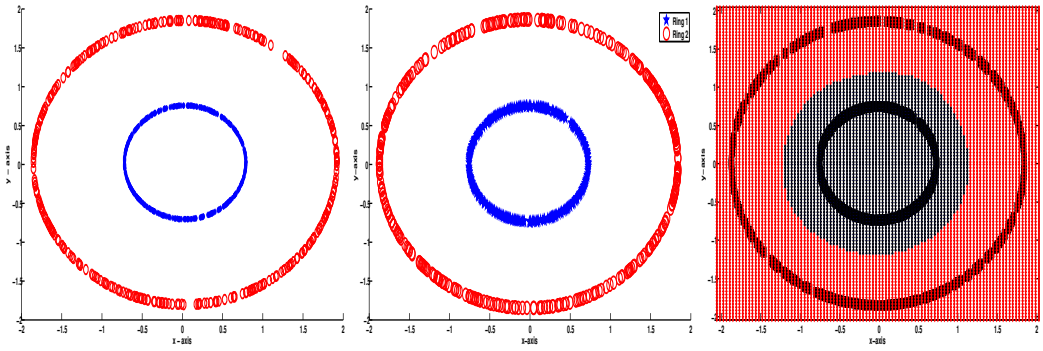


Figure 2: Ring data set (1st plot), KGLVQ model (2nd plot), the outer ring is shown in red and using 'o' while the inner ring is plotted in blue $\star$. The 3rd plot shows the cluster boundaries of the model from the 2nd plot. The model was calculated without sparsity. It can be clearly seen that the AKGLVQ with an rbf kernel successfully separated these two clusters and also the cluster boundaries are very well approximated with a large margin between the two rings.

All AKGLVQ and KGLVQ models are obtained with 1 prototype per class, using $C = 100$ cycles and with a nyström approximation of $q = 0.1 \times N$ of the original kernel matrix for the AKGLVQ variant, sparsity was switched *off*. The value of the nyström approximation is not so critical but should be not lower than 10% to keep sufficient approximation accuracy. It has mainly an influence on the runtime performance as long as the data space is sufficiently densely sampled.

	Dim	Size	KGLVQ	#PT	AKGLVQ	#PT	SVM	#SV
Breast Cancer	32	569	92.97±01.87	2	92.27±03.43	2	97.71±01.45	512
Diabetes	8	768	71.88±04.79	2	71.56±06.19	2	76.42±04.20	691
Heart	13	270	81.85±05.91	2	81.11±06.16	2	84.07±08.38	243

Table 1: Generalization accuracy and model complexity (averaged) for the datasets. The AKGLVQ makes use of the Nyström approximation with 10% of the distances, sparsity has been switched off. The memory used to store the kernel matrix for AKGLVQ is $\approx 95\%$ less then for KGLVQ and a speedup of 2 to 7 could be observed in average. The generalization of AKGLVQ is almost the same like for KGLVQ and is also quite good compared to SVM. #PT refers to the number of prototypes, whereas #SV provides the number of support vectors in the final model.

The results of AKGLVQ compare favorable in comparison to KLVQ or SVM but especially the runtime is significantly improved with respect to KGLVQ see Table 1. We find that the prediction performance of AKGLVQ and KGLVQ are quite similar, and both are competitive to SVM. KGLVQ however is not really applicable for larger data sets due to the costly distance calculations using the full kernel.

4.3 Complexity and runtime analysis

The original KGLVQ algorithm employs a full, quadratic kernel matrix in the distance calculations and is optimizing the $M \times N$ coefficient matrix Ψ. The selected underlying iterative optimization scheme is gradient descend. The optimization is done for C cycles

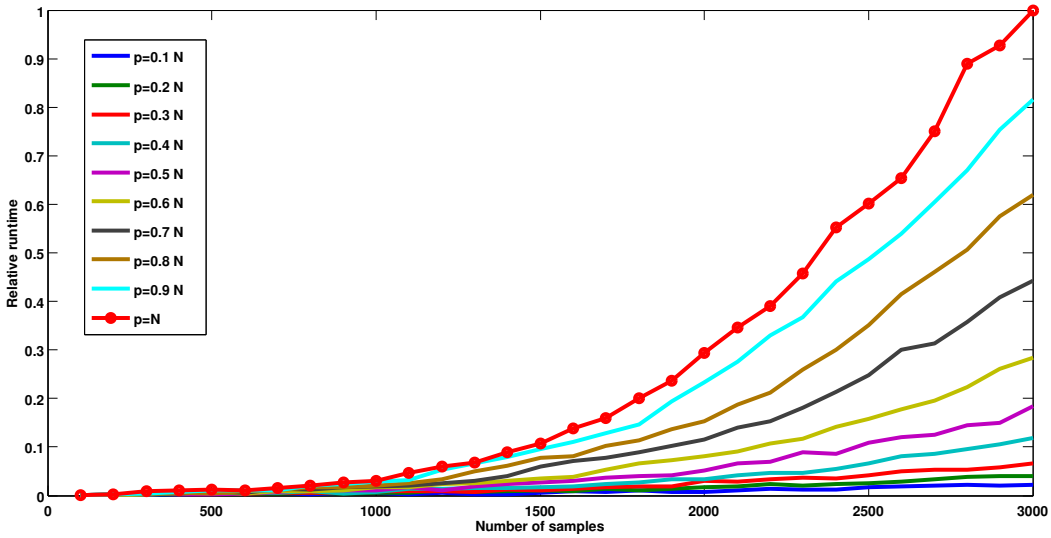


Figure 3: Runtime analysis of AKGLVQ using an extended ring data set for different values of q in the Nyström approximation. The number of samples is changed within $100 - 3000$ on the x-axis, with the relative runtime given on y. The different curves are obtained by changing the Nyström approximation from 10% - 100%. The KGLVQ curve is given with o on the sampling points.

as an upper limit and often independent of the data chosen as $C = 100$. The number of prototypes is typically chosen independent to the real data set size and in general much smaller than N, such that the memory complexity of Ψ is linear in $O(M \times N)$. Taking this into account KGLVQ has a memory complexity of roughly $O(N^2)$. Each distance calculation involves matrix/vector operations with a N^2 matrix which has to be done for all N data point and for C cycles. Hence the runtime complexity is in the range of $O(N^3)$.

The AKGLVQ algorithm provides two approaches to optimize runtime and memory complexity, namely the Nyström approximation and the sparsity constraint as pointed out before. Using the Nyström approximation the memory complexity of the kernel matrix is reduced to $O(q \times N)$. Hence the necessary memory to store the kernel matrix as well as the number of matrix operation is directly reduced depending on q. For most data sets it is reasonable to set q to a small fraction of N e.g. 10%. The memory complexity of the matrix Ψ is unaffected. This leads to an estimated linear memory consumption of $O((q + M) \times N)$. To obtain the two matrices of the Nyström approximation a (pseudo) inverse has to be calculated and for the Nyström based distance calculation additional multiplications by a $q \times N$ matrix from both sides are necessary. This leads to a linear runtime complexity of $O(q^3 + qN)$. The full runtime complexity of AKGLVQ is however quadratic because the operations have to be done for all N data points, so we finally get a quadratic setting of $O(N^2)$. A runtime analysis of the *ring* data set with a maximum number of 3000 point is depicted in Figure 3.

By employing the new sparsity measure it is also possible to reduce the complexity of the model and to reduce the amount of memory necessary to store Ψ. The associated parameter β can be estimated by a cross-validation scheme on a sub set of the data using a grid search within a reasonable range of $[0, \ldots, 50]$. In Figure 4 the effect of the sparsity approach with respect to prediction accuracy on a test set and the time complexity is shown. The accuracy and memory complexity is given in % whereas for the time complexity the maximal necessary time is normalized to 1 to allow for better

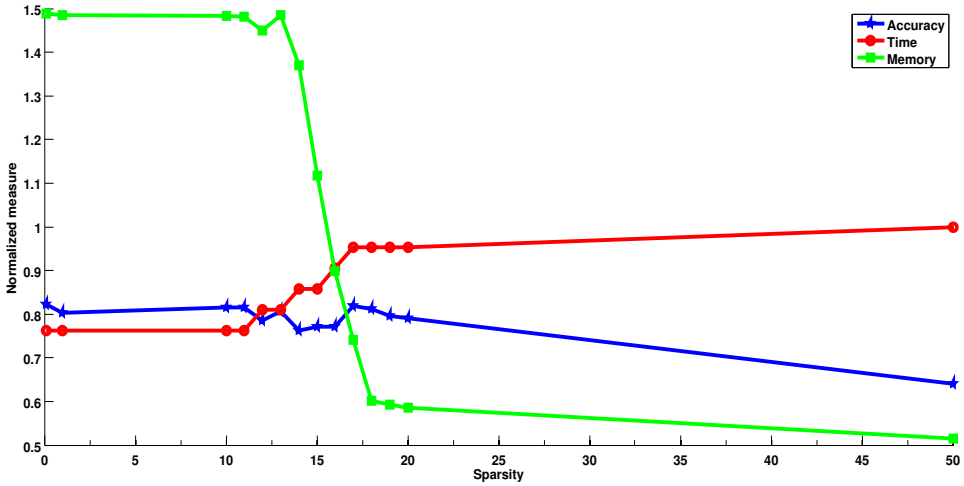


Figure 4: Complexity analysis of AKGLVQ using the sparsity constraint for the heart data set (profiles are similar for the other data sets). The measures are calculated from the observed training model, the memory consumption refers to the matrix Ψ only.

comparison. Using the sparsity constraint and sparse matrices the initial amount of necessary memory is higher than without sparsity due to the overhead caused by the management of sparse matrices. The sparsity constraint is not a hard control parameter for the memory complexity of the model hence it is not possible to provide theoretical guarantees of the memory consumption.

Analyzing Figure 4 we observe that the prediction accuracy is smoothly decreasing with increased β. The optimal β value is around $\beta = 19$ with around 79% accuracy and 40% less consumed memory. An analysis of the other data sets showed that, as expected, the β parameter is data set specific. It should be optimized based on an independent test set and used to balance accuracy and model complexity.

As an overall observation we found that the runtime of the algorithm is increased by around 20% using the sparsity measure due to additional normalization steps and the effort to manage the sparse matrices. Considering the prior analysis it is obvious that the sparsity constraint can not directly used to speedup the learning time or to reduce (continuously) the consumed memory. Instead it should be used to simplify the final model. This is especially relevant if we focus on interpretable models and a small number of non-vanishing coefficients provides easier access to the interpretation of the prototypes, analyzing the linear combination.

4.4 Medium sample size data

In a further study we analyze the accelerated KGLVQ and SVM on medium and large data sets with multiple thousand samples. Thereby we make use of the UCI *spam*-data set[4] which contains measurements to predict if a obtained email is to classify as spam. Further we use the *usps*-data set, containing 16×16 gray-scale images of handwritten digits, provided in[15] and the CMU *faces*-data from the UCI database [4], which contains 640 b/w images of people taken with varying pose, expression, eyes presentation and size. The later two are multiclass data sets. The standard KGLVQ can not any longer be applied for these data under valid settings, without significant sub-sample selections on the training data, which has a negative impact on the results. For the USPS data we took a commonly used subset of 2000 samples randomly sampled. All results are obtained in a 10-fold cross validation using the ELM kernel with 100 cycles

for AKGLVQ. Multiclass classification for SVM was done using a 1 vs rest scheme. Results are shown in Table 2.

	#C	Dim	Size	AKGLVQ	AKGLVQ-Compl.		SVM	SVM-Compl.	
					ø-T	#PT		ø-T	ø-#SV
Spam	2	57	4601	86.57 ± 02.64	130.73	0.43% (2)	91.92 ± 02.01	00.56	33.13% (469)
USPS	10	256	2000	81.70 ± 01.55	18.95	2.22% (40)	91.35 ± 02.46	00.32	100% (1800)
Faces	20	15360	640	81.09 ± 06.39	01.20	3.47% (20)	94.84 ± 03.83	00.67	10.95% (63)

Table 2: Generalization accuracy and model complexity (averaged) for the small datasets. $\emptyset - T$ refers to the mean runtime in minutes, #PT denotes the number of prototypes and #SV the number of support vectors, respectively. Note that the complexity values are calculated with respect to the training data.

We observe that the AKGLVQ was quite efficient in modeling the given problems but the prediction performance is significantly lower than the one obtained by SVM also the overall runtime is worse than that of SVM which is typically a magnitude faster than AKGVLQ. However we would like to point out again that our objective is to improve kernel based prototype methods, namely KGLVQ rather to compete directly with SVM. The most interesting property of prototype classifiers may not be the prediction accuracy, although it is quite good in general, but more the interpretability and other aspect as shown in the following.

The higher runtime complexity of AKGVLQ is expected because it is an online learning algorithm in contrast to SVM. AKGLVQ still scales quadratic as pointed out before, if no additional techniques like active learning [30] are employed, which however does not provide guarantees. On the other hand this also allows an easy retraining in case of novel data which is not directly accessible using SVM approaches. Interestingly the quite good prediction results of AKGLVQ are already obtained with very few prototypes leading to compact models. An increase of the number of prototypes up-to a factor of 10 does not change the prediction accuracy significantly. SVM however has used at least 10% of the data or like for the USPS data the whole training data set, making the model very complex.

The KGLVQ and its approximated variant are prototype based methods and the prototypes are constructed by means of a linear combination of the data points in the coefficient matrix Ψ. By analyzing the coefficient matrix Ψ of the models shown in Table 2, it is possible to identify items from the original data set which are considered to be most characteristic and important for the voronoi cell generated by a specific prototype, this is in contrast to SVM models because their model parameters are extreme points rather prototypes. For the USPS dataset which consists of digit images and the faces data set with images of faces it is possible to obtain direct reconstructions of the prototypes which can be easily visualized and interpreted. In Figure 6 the visualizations of the digits $'0', '2', '8'$ are shown using either the median of the class, the prototype reconstruction or the median support vector reconstruction for support vectors of the corresponding class.

Figure 5 shows different digits of the USPS data set [5]. The Figure has been regenerated as described in [5] for the public available USPS data set using the NeXOM algorithm, which is a specific variant of neighborhood embedding, comparable e.g. to Multi-Dimensional Scaling (MDS)[15][3]. Analyzing the USPS data for the digits $'2'$ and $'8'$ we find that the crossings of the arcs are quite well defined independent of the specific writer but the position of the arcs in the outer regions of the digit differ. This is reflected by the median reconstructions, plots (a) which show holes for these regions. In contrast the learned prototypes for $'2'$ and $'8'$ do not suffer from this error but show

[3]The picture is not identical but similar due to random effects in the initialization of NeXOM

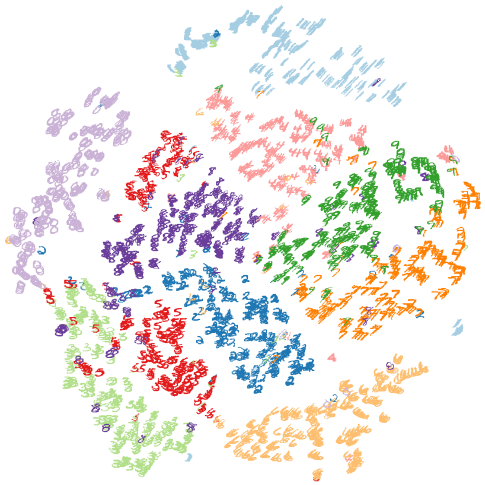


Figure 5: Different USPS symbols in a two dimensional projection using the NeXOM algorithm.

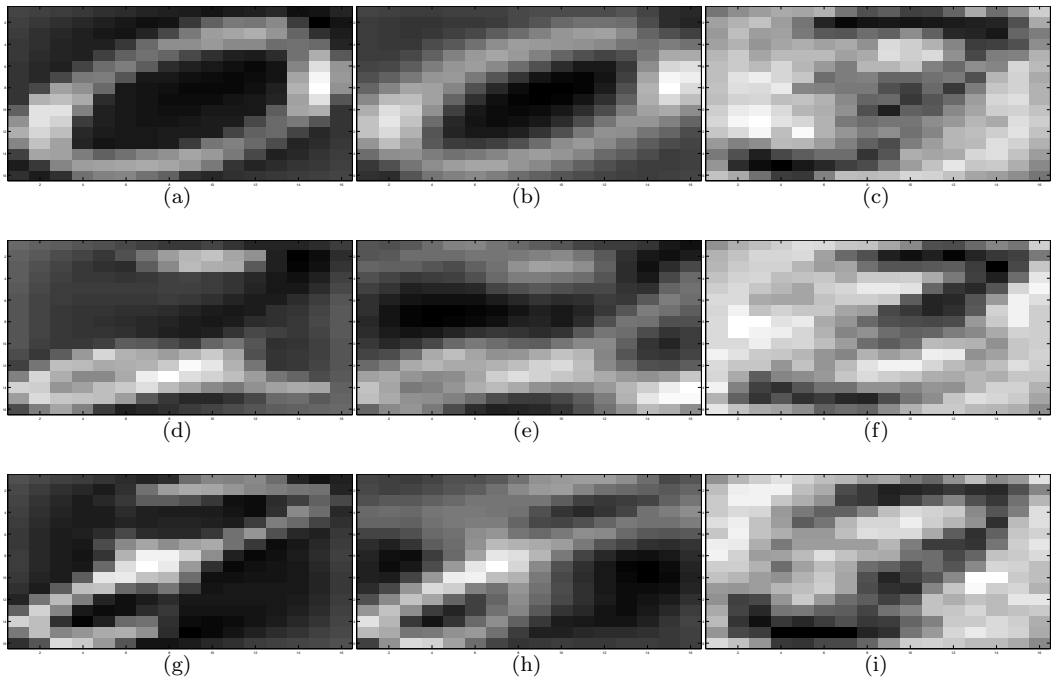
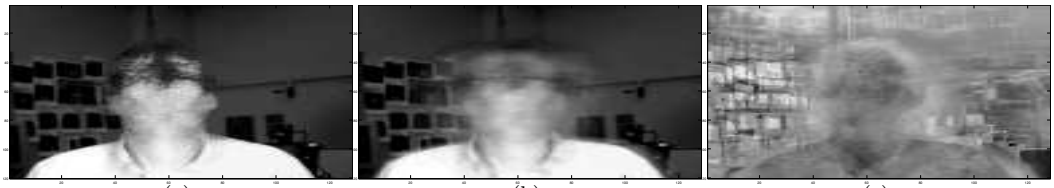


Figure 6: Reconstruction of digit representants. The first row shows reconstructions of the digit $'0'$, the second of the digit $'2'$ and the last of the digit $'8'$. The plot (a) is always the median image of the corresponding class, (b) the learned prototype representation, (c) the support vector reconstruction.



(a) (b) (c)

Figure 7: Reconstruction of faces representants for one of the faces classes. The plot (a) is the median image of the corresponding class, (b) the learned prototype representation, (c) the support vector reconstruction.

a more realistic, prototypical representation of the digit. For the SVM we would expect that the items at the decision borders are pronounced most, such that the most atypical items are represented. Indeed the plots (c) appear to be quite blurred and are hard to interpret. For the digit $'0'$ one observes that the median and prototype reconstruction are almost identical, which is caused by a strong homogeneity in the data. The rare abnormal $'0's$ in the data are either wrong oriented, with an open loop, very tight or almost without the hole, appearing as a bold blot. These examples are selected by the SVM as the model parameters and hence the reconstruction in (c) is hard to interpret as well[4].

For the faces data, exemplary results are shown in Figure 7. The shown person moved the head in the different recordings, so more or less only the background and the body shape are stable. This is reflected by all reconstructions. The median plot (a) shows the raw shape of the person but the face is blurred. The prototype reconstruction (b) is less blurred and reflects also the movement of the head, showing also traces of the turned heads in the image. The prototype is actually so accurate that also the sun-glasses and the lips, nose and ears can be identified. The reconstruction of the SVM (c) is again hard to interpret. The raw shape is preserved but the shown picture is clearly not representative but more a mixture of the abnormal cases in the data, as expected.

Considering the model complexity we find that with very few prototypes for all datasets the AKGLVQ performs quite well. Using the Nyström approximation the memory consumption of the kernel matrix can be substantial reduced such that AKGLVQ becomes an interesting, efficient and prototype based complement to SVM. This is especially interesting if an interpretable prototype of a class is needed like for image retrieval systems to label the underlying data by a typical image. Also in other cases of interpretable data like clinical recordings, the prototypical model parameters are much easier accessible for the domain expert. It also helps to get a better understanding of the information encoded in the model. If the model fails to classify specific items in the data or assigns them constantly to one wrong class, the prototype can help to interpret the reason for this. In that way the system can be improved by incorporating additional knowledge in an user interactive manner.

Overall we found that AKGLVQ is now capable to learn also very complex data sets with multiple 1000 of items and due to the Nyström approximation and the integrated sparsity constraint the memory complexity of the model is quite low. AKGLVQ allows, like all prototype methods, explicit control over the model complexity by specifying the number of prototypes. For the considered data the prediction accuracy of AKGLVQ was similar to that of KGLVQ but with a significant reduced model complexity and a substantial speed up in the calculations. In comparison to SVM the AKGLVQ mod-

[4]The SVM model of the USPS data contains all points, because all α-weights are significant different from 0, but the extreme symbols have the largest α weights.

els are very compact but were less efficient in prediction for the large data set while comparable effective for the experiments with the more simple data. The obtained prototypes of the KGLVQ- and AKGLVQ-model are much easier to interpret, whereas for SVM the model is less informative.

5 CONCLUSIONS

In this paper we proposed an extended variant of kernelized learning vector quantizer with a significantly reduced model complexity through the integration of the Nyström method and sparse learning. The obtained models use much less memory due to a compact, approximated kernel representation and a sparse coefficient matrix Ψ. Further we compared the efficiency of our new approach with KGLVQ and SVM considering prediction accuracy, model complexity and interpretability. We found that the generalization capability of AKGLVQ is similar to those of KGLVQ and less to SVM. AKGLVQ is much quicker than KGLVQ and needs markable less memory. AKGLVQ and KGLVQ provides interpretable models in contrast to SVM. If not only prediction accuracy but also compactness and interpretability matters AKGLVQ provide an interesting alternative to the considered standard kernel learning methods and is now applicable for medium-sized sets of data, which was not possible before. One very important subject of future works will be to further decrease runtime and memory requirements while parallel increase the prediction efficiency for extremely large data sets.
Acknowledgment: This work was supported by the German Res. Fund. (DFG), HA2719/4-1 (Relevance Learning for Temporal Neural Maps) and by the Cluster of Excellence 277 Cognitive Interaction Technology funded in the framework of the German Excellence Initiative. The authors would like to thank Kerstin Bunte, University of Groningen, The Netherlands, for providing the USPS visualization.

References

[1] Wesam Barbakh and Colin Fyfe. Online clustering algorithms. *Int. J. Neural Syst.*, 18(3):185–194, 2008.

[2] P. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2003.

[3] C.M. Bishop. *Pattern Recognition and Machine Learning.* Springer Science+Business Media, LLC, New York, NY, 2006.

[4] C.L. Blake and C.J. Merz. UCI repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science, available at: http://www.ics.uci.edu/ mlearn/MLRepository.html, 1998.

[5] Kerstin Bunte, Barbara Hammer, Thomas Villmann, Michael Biehl, and Axel Wismüller. Neighbor embedding XOM for dimension reduction and visualization. *Neurocomputing*, 74(9):1340–1350, 2011.

[6] Emilio Corchado and Colin Fyfe. Relevance and kernel self-organising maps. In Okyay Kaynak, Ethem Alpaydin, Erkki Oja, and Lei Xu, editors, *ICANN*, volume 2714 of *Lecture Notes in Computer Science*, pages 280–290. Springer, 2003.

[7] K. Crammer, R. Gilad-Bachrach, A.Navot, and A.Tishby. Margin analysis of the LVQ algorithm. In *Proc. NIPS 2002*, pages 462–469, 2002.

[8] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* Cambridge University Press, 2010.

[9] B. Frenay and M. Verleysen. Parameter-free kernel in extreme learning for non-linear support vector regression. *NeuroComputing*, page in press, 2010.

[10] Roberto Gil-Pita and Xin Yao. Evolving edited k-nearest neighbor classifiers. *Int. J. Neural Syst.*, 18(6):459–467, 2008.

[11] B. Hammer, M. Strickert, and Th. Villmann. Relevance LVQ versus SVM. In L. Rutkowski, J. Siekmann, R. Tadeusiewicz, and L.A. Zadeh, editors, *Artificial Intelligence and Soft Computing (ICAISC 2004)*, Lecture Notes in Artificial Intelligence 3070, pages 592–597. Springer Verlag, Berlin-Heidelberg, 2004.

[12] B. Hammer and Th. Villmann. Generalized relevance learning vector quantization. *Neural Networks*, 15(8-9):1059–1068, 2002.

[13] B. Hammer and Th. Villmann. Mathematical aspects of neural networks. In M. Verleysen, editor, *Proc. Of European Symposium on Artificial Neural Networks (ESANN'2003)*, pages 59–72, Brussels, Belgium, 2003. d-side.

[14] Barbara Hammer, Marc Strickert, and Thomas Villmann. On the generalization ability of GRLVQ networks. *Neural Processing Letters*, 21(2):109–120, 2005.

[15] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, New York, 2001.

[16] P. Hoyer. Non-negative Matrix Factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[17] A. K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31:651–666, 2010.

[18] Alexander Kaiser, Wolfram Schenck, and Ralf Möller. Coupled singular value decomposition of a cross-covariance matrix. *Int. J. Neural Syst.*, 20(4):293–318, 2010.

[19] S. Sathiya Keerthi, Olivier Chapelle, and Dennis DeCoste. Building support vector machines with reduced classifier complexity. *Journal of Machine Learning Research*, 7:1493–1515, 2006.

[20] Teuvo Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 1995. (Second Extended Edition 1997).

[21] K. Labusch, E. Barth, and T. Martinetz. Learning data representations with sparse coding neural gas. In *Proc. of ESANN'08*, pages 233–238, 2008.

[22] Mu. Li, J.T. Kwok, and B.-L. Lu. Making large-scale Nyström approximation possible. In *Proceedings of the International Conference on Machine Learning (ICML)'2010*, 2009.

[23] Ezequiel López-Rubio, Rafael Marcos Luque Baena, and Enrique Domínguez. Foreground detection in video sequences with probabilistic self-organizing maps. *Int. J. Neural Syst.*, 21(3):225–246, 2011.

[24] Joshua E. Menke and Tony R. Martinez. Improving supervised learning by adapting the problem to the learner. *Int. J. Neural Syst.*, 19(1):1–9, 2009.

[25] Britta Mersch, Tobias Glasmachers, Peter Meinicke, and Christian Igel. Evolutionary optimization of sequence kernels for detection of bacterial gene starts. *Int. J. Neural Syst.*, 17(5):369–381, 2007.

[26] B. Olshausen and D. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Letters to Nature*, 381:607–609, 1996.

[27] John C. Platt. Fast training of support vector machines using sequential minimal optimization. pages 185–208. MIT Press, Cambridge, MA, USA, 1999.

[28] A. K. Qin and P. N. Suganthan. A novel kernel prototype-based learning algorithm. In *Proc. of ICPR'04*, pages 2621–624, 2004.

[29] A. S. Sato and K. Yamada. Generalized learning vector quantization. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 423–429. MIT Press, 1995.

[30] F.-M. Schleif, B. Hammer, and Th. Villmann. Margin based active learning for LVQ networks. *Neurocomputing*, 70(7-9):1215–1224, 2007.

[31] F.-M. Schleif, T. Villmann, B. Hammer, P. Schneider, and M. Biehl. Generalized derivative based kernelized learning vector quantization. In *Proceedings of IDEAL 2010*, pages 21–28, 2010.

[32] P. Schneider, M. Biehl, and B. Hammer. Adaptive relevance matrices in learning vector quantization. *Neural Computation*, 21:3532–3561, 2009.

[33] B. Schoelkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

[34] S. Seo and K. Obermayer. Soft learning vector quantization. *Neural Computation*, 15:1589–1604, 2003.

[35] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis and Discovery*. Cambridge University Press, 2004.

[36] Liangdong Shi, Yinghuan Shi, Yang Gao, Lin Shang, and Yubin Yang. XCSC: a novel approach to clustering with extended classifier system. *Int. J. Neural Syst.*, 21(1):79–93, 2011.

[37] Ivor Wai-Hung Tsang, András Kocsor, and James Tin-Yau Kwok. Large-scale maximum margin discriminant analysis using core vector machines. *IEEE Transactions on Neural Networks*, 19(4):610–624, 2008.

[38] V Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[39] C. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, pages 682–688. 2001.

[40] Yang Yang and Bao-Liang Lu. Protein subcellular multi-localization prediction using a min-max modular support vector machine. *Int. J. Neural Syst.*, 20(1):13–28, 2010.

[41] K. Zhang, I.W. Tsang, and J.T. Kwok. Improved Nyström low-rank approximation and error analysis o. In *Proceedings of the International Conference on Machine Learning (ICML)'2010*, 2009.

4.2 Linear time relational prototype based learning

The article *Linear time relational prototype based learning* by A. Gisbrecht, B. Mokbel, F.M.**Schleif** and X. Zhu and B. Hammer appeared in the Journal of Neural Systems 22(5), online 2012 Generalized Learning Vector Quantizers (GLVQ) are extended by relational learning and coupled with the Nyström approximation to keep linear learning complexity also for very large data sets. All authors contributed equally to the article. A. Gisbrecht provided initial work on Nyström approximations for dissimilarity data and an implementation of the relational GTM, I derived and implemented GLVQ for relational learning and integrated the Nyström approximation. B. Mokbel and X. Zhu run the experiments and prepared the data. B. Hammer supervised the project and all authors discussed the general article.

Additional publications in international conferences and journal publications where I am co-author and which cover a similar or related topic include:

1. F.-M. **Schleif** and A. Gisbrecht, *Data analysis of (non-)metric proximities at linear costs*, Proceedings of SIMBAD 2013, accepted, 2013 (Content: Our main contribution is an efficient *linear* technique, to convert (potentially non-metric) large scale dissimilarity matrices into approximated psd kernel matrices.)

2. B. Hammer, D. Hofmann, F.-M. **Schleif**, X. Zhu, *Learning vector quantization for (dis-)similarities*, NeuroComputing, accepted, 2013 (Content: Proposal of a general framework for dissimilarity based learning, including kernel generalized relevance LVQ, relational generalized relevance LVQ, kernel robust soft LVQ and relational robust soft LVQ. Also unsupervised prototype based techniques which are based on a cost function can put into this framework.)

3. F.-M. **Schleif**, X. Zhu, A. Gisbrecht, B. Hammer, *Fast approximated relational and kernel clustering*, In Proceedings of the International Conference on Pattern Recognition ICPR 2012, 1229-1232, 2012 (Content: The approach combines similarity and dissimilarity learning for very large datasets in a common framework. Additionally a fast batch kernel prototype classifier is proposed.)

4. F.-M. **Schleif**, X. Zhu, B. Hammer, *Soft Competitive Learning for large data sets*, In Proceedings of ADBIS 2012, 141-151, 2012 (Content: The article presents a core based approach of soft competitive learning, as a novel clustering algorithm for large scale problems)

5. X. Zhu, A. Gisbrecht, F. - M. **Schleif**, B. Hammer, *Approximation techniques for clustering dissimilarity data*, NeuroComputing 90, 72-84, 2012. (Content: The article presents an unsupervised patch-processing approach for relational data learning)

6. X. Zhu, F.-M. Schleif and Barbara Hammer, *Patch Processing for Relational Learning Vector Quantization*, In Proceedings of ISNN, 2012, 55-63, 2012. (Content: A patch strategy is introduced for *supervised* proximity learning)

7. B. Hammer, B. Mokbel, F.-M. **Schleif**, X. Zhu, *Prototype-Based Classification of Dissimilarity Data*, In Proceedings of Intelligent Data Analysis (IDA)'2011, 185-197, 2011 (Content: Prototype-based classification is extended towards general dissimilarities)

Linear time relational prototype based learning

Andrej Gisbrecht
Dept. of Techn., Univ. of Bielefeld, Universitätsstrasse 21-23
33615 Bielefeld, Germany
E-mail: agisbrec@techfak.uni-bielefeld.de

Bassam Mokbel
Dept. of Techn., Univ. of Bielefeld, Universitätsstrasse 21-23
33615 Bielefeld, Germany
E-mail: bmokbel@techfak.uni-bielefeld.de

Frank-Michael Schleif
Dept. of Techn., Univ. of Bielefeld, Universitätsstrasse 21-23
33615 Bielefeld, Germany
E-mail: fschleif@techfak.uni-bielefeld.de

Xibin Zhu
Dept. of Techn., Univ. of Bielefeld, Universitätsstrasse 21-23
33615 Bielefeld, Germany
E-mail: xzhu@techfak.uni-bielefeld.de

Barbara Hammer[1]
Dept. of Techn., Univ. of Bielefeld, Universitätsstrasse 21-23
33615 Bielefeld, Germany
E-mail: bhammer@techfak.uni-bielefeld.de

Abstract

Prototype based learning offers an intuitive interface to inspect large quantities of electronic data in supervised or unsupervised settings. Recently, many techniques have been extended to data described by general dissimilarities rather than Euclidean vectors, so-called relational data settings. Unlike the Euclidean counterparts, the techniques have quadratic time complexity due to the underlying quadratic dissimilarity matrix. Thus, they are infeasible already for medium sized data sets. The contribution of this article is twofold: on the one hand we propose a novel supervised prototype based classification technique for dissimilarity data based on popular learning vector quantization, on the other hand we transfer a linear time approximation technique, the Nyström approximation, to this algorithm and an unsupervised counterpart, the relational generative topographic mapping. This way, linear time and space methods result. We evaluate the techniques on three examples from the biomedical domain.

[1]corresponding author

1 INTRODUCTION

In many application areas such as bioinformatics, technical systems, or the web, electronic data sets are increasing rapidly with respect to size and complexity. Machine learning has revolutionized the possibility to deal with large electronic data sets in these areas by offering powerful tools to automatically extract a regularity from given data. Popular approaches provide diverse techniques for data structuring and data inspection. Visualization, clustering, or classification still constitute one of the most common tasks in this context [3, 12, 37, 30].

Topographic mapping such as offered by the self-organizing map (SOM) [18] and its statistic counterpart, the generative topographic mapping (GTM) [6] provide simultaneous clustering and data visualization. For this reason, topographic mapping constitutes a popular tool in diverse areas ranging from remote sensing or biomedical domains up to robotics or telecommunication [18, 21]. As an alternative, learning vector quantization (LVQ) represents priorly given classes in terms of labeled prototypes [18]. Learning typically takes place by means of Hebbian and anti-Hebbian updates. Original LVQ is based on heuristic grounds while modern alternatives are typically derived from an underlying cost function [33]. Similar to its unsupervised counterpart, LVQ has been successfully applied in diverse areas including telecommunication, robotics, or biomedical data analysis [18, 2].

Like many classical machine learning techniques, GTM and LVQ have been proposed for Euclidean vectorial data. Modern data are often associated to dedicated structures which make a representation in terms of Euclidean vectors difficult: biological sequence data, text files, XML data, trees, graphs, or time series, for example [31, 34]. These data are inherently compositional and a feature representation leads to information loss. As an alternative, a dedicated dissimilarity measure such as pairwise alignment, or kernels for structures can be used as the interface to the data In such cases, machine learning techniques which can deal with pairwise similarities or dissimilarities have to be used [24].

Also kernel methods like the Support Vector Machine (SVM) (see e.g.[9]) can be used for dissimilarity data, but complex preprocessing steps are necessary as discussed in the following. Kernel methods are known to be very effective, with respect to the generalization ability and, using modern approximation schemes, are also reasonable effective for larger data sets. In contrast to prototype methods the cost function is formulated typically by means of a convex problem, such that standard and effective optimization techniques can be used. Often they automatically adapt the model complexity, e.g. by means of support vectors for SVM, in accordance to the given *supervised* problem, which is often not the case for prototype methods. This strong framework however, requires a valid positive semi-definite kernel as an input, which is often not directly available for dissimilarity data. In fact, as discussed in the work of Pekalska[25], dissimilarity data can encode information in the euclidean and non-euclidean space and transformations to obtain a valid kernel may be inappropriate[32].

Quite a few extensions of prototype-based learning towards pairwise similarities or dissimilarities have been proposed in the literature. Some are based on a kernelization of existing approaches [7, 39, 29], while others restrict the setting to exemplar based techniques [10, 19]. Some techniques build on alternative cost functions and advanced optimization methods [35, 15]. A very intuitive method which directly extends prototype based clustering to dissimilarity data has been proposed in the context of fuzzy clustering [17] and later been extended to topographic mapping such as SOM and GTM [16, 14]. Due to its direct correspondence to standard topographic mapping in the Euclidean case, we will focus on the latter approach. We will exemplarily look at this relational extension of GTM to investigate the performance of unsupervised prototype-based techniques for dissimilarity data. In this contribution, we will propose, as an alternative, a novel supervised prototype based classification scheme for dissimilarity data, with initial work given in [28]. Essentially, a modern LVQ formulation which is based on a cost function will be extended using the same trick to assess relational data.

One drawback of machine learning techniques for dissimilarities is given by their high computational costs: since they depend on the full (quadratic) dissimilarity matrix, they have squared time complexity; further, they require the availability of the full dissimilarity matrix, which is even the more severe bottleneck if complex dissimilarities such as e.g. alignment techniques are used. This fact makes the methods unsuitable already for medium sized data sets.

Here, we propose a popular approximation technique to speed up prototype based methods for dissimilarities: the Nyström approximation has been proposed in the context of kernel methods as a low rank approximation of the matrix [38]. In [13], preliminary work extends these results to dissimilarities. In this contribution, we demonstrate that the technique provides a suitable linear time approximation for GTM and LVQ for dissimilarities.

Now we first shortly recall the classical GTM and a variant of LVQ. Then we introduce the general concept underlying relational data representation, and we transfer this principle to GTM (shortly summarizing the results already presented in [14]) and to LVQ. The latter gives the novel algorithm relational generalized learning vector quantization. We recall the derivation of the low rank Nyström approximation for similarities and transfer this principle to dissimilarities. Linear time techniques for relational GTM and relational LVQ result. We demonstrate the behavior of the techniques in applications from the biomedical domain.

2 TOPOGRAPHIC MAPPING

Generative Topographic Mapping (GTM) has been proposed in [6] as a probabilistic counterpart to SOM. It models given data $\mathbf{x}^i \in \mathbb{R}^n$ by a constraint mixture of Gaussians induced by a low dimensional latent space. More precisely, regular lattice points $\mathbf{w}$ are fixed in latent space and mapped to target vectors $\mathbf{w} \mapsto \mathbf{t} = y(\mathbf{w}, \mathbf{W})$ in the data space, where the function y is typically chosen as generalized linear regression model $y : \mathbf{w} \mapsto \Phi(\mathbf{w}) \cdot \mathbf{W}$. The base functions Φ could be chosen as any set of nonlinear functions. Typically, equally spaced Gaussians with bandwidth σ are taken.

These prototypes in data space give rise to a constraint mixture of Gaussians in the following way. Every latent point induces a Gaussian

$$p(\mathbf{x}|\mathbf{w}, \mathbf{W}, \beta) = \left(\frac{\beta}{2\pi}\right)^{\frac{n}{2}} \exp\left(-\frac{\beta}{2}\|\mathbf{x} - y(\mathbf{w}, \mathbf{W})\|^2\right) \tag{1}$$

A mixture of K modes $p(\mathbf{x}|\mathbf{W}, \beta) = \sum_{k=1}^{K} \frac{1}{K} p(\mathbf{x}|\mathbf{w}^k, \mathbf{W}, \beta)$ is generated. GTM training optimizes the data log-likelihood with respect to $\mathbf{W}$ and β. This can be done by an EM approach, iteratively computing responsibilities

$$R_{ki}(\mathbf{W}, \beta) = p(\mathbf{w}^k|\mathbf{x}^i, \mathbf{W}, \beta) = \frac{p(\mathbf{x}^i|\mathbf{w}^k, \mathbf{W}, \beta)}{\sum_{k'} p(\mathbf{x}^i|\mathbf{w}^{k'}, \mathbf{W}, \beta)} \tag{2}$$

of component k for point $\mathbf{x}^i$, and optimizing model parameters by means of the formulas

$$\mathbf{\Phi}^T \mathbf{G}_{\text{old}} \mathbf{\Phi} \mathbf{W}_{\text{new}}^T = \mathbf{\Phi}^T \mathbf{R}_{\text{old}} \mathbf{X} \tag{3}$$

for $\mathbf{W}$, where $\mathbf{\Phi}$ refers to the matrix of base functions Φ evaluated at the lattice points $\mathbf{w}^k$, $\mathbf{X}$ refers to the data points, $\mathbf{R}$ to the responsibilities, and $\mathbf{G}$ is a diagonal matrix with accumulated responsibilities $G_{ii} = \sum_i R_{ki}(\mathbf{W}, \beta)$. The bandwidth is given by

$$\frac{1}{\beta_{\text{new}}} = \frac{1}{ND} \sum_{k,i} R_{ki}(\mathbf{W}_{\text{old}}, \beta_{\text{old}}) \|\Phi(\mathbf{w}^k)\mathbf{W}_{\text{new}} - \mathbf{x}^i\|^2 \tag{4}$$

where D is the data dimensionality and N the number of points. GTM is initialized by aligning the lattice image and the first two data principal components.

3 LEARNING VECTOR QUANTIZATION

As before, data $\mathbf{x}^i \in \mathbb{R}^n$ are given. Here we consider the crisp setting. That means, prototypes $\mathbf{w}^j \in \mathbb{R}^n, j = 1, \ldots, K$ in the data space decompose data into receptive fields $R(\mathbf{w}^j) := \{\mathbf{x}^i \ : \ \forall k \ d(\mathbf{x}^i, \mathbf{w}^j) \leq d(\mathbf{x}^i, \mathbf{w}^k)\}$ based on the squared Euclidean distance $d(\mathbf{x}^i, \mathbf{w}^j) = \|\mathbf{x}^i - \mathbf{w}^j\|^2$.

For supervised learning, data $\mathbf{x}^i$ are equipped with class labels $c(\mathbf{x}^i) \in \{1, \ldots, L\} = \mathcal{L}$. Similarly, every prototype is equipped with a priorly fixed label $c(\mathbf{w}^j)$. Let $\mathbb{W}_c = \left\{\mathbf{w}^l | c(\mathbf{w}^l) = c\right\}$ be the subset of prototypes assigned to class $c \in \mathcal{L}$. A data point is classified according to the class of its closest prototype. The classification error of this mapping is given by the term $\sum_j \sum_{\mathbf{x}^i \in R(\mathbf{w}^j)} \delta(c(\mathbf{x}^i) \neq c(\mathbf{w}^j))$ with the delta function δ. This cost function cannot easily be optimized explicitly due to vanishing gradients and discontinuities. Therefore, LVQ relies on a reasonable heuristic by performing Hebbian updates of the prototypes, given a data point [18]. Recent alternatives derive similar update rules from explicit cost functions which are related to the classification error, but display better numerical properties such that efficient optimization algorithms can be derived thereof [33, 26, 36].

We introduce two special notations for the prototype which is closest to a given point $\mathbf{x}^i$ with the same label: $\mathbf{w}^+$ or a different label: $\mathbf{w}^-$. The corresponding distance d_i^+, d_i^-:

$$\begin{aligned} d_i^+ &= d(\mathbf{w}^+, \mathbf{x}^i) \text{ with } \mathbf{w}^+ \in \mathbb{W}_c, \ c = c(\mathbf{x}^i), \\ \mathbf{w}^+ &:= \mathbf{w}^l : d(x^i, w^l) \leq d(x^i, w^j), \{\mathbf{w}^j, \mathbf{w}^l\} \in \mathbb{W}_c \\ d_i^- &= d(\mathbf{w}^-, \mathbf{x}^i) \text{ with } \mathbf{w}^- \notin \mathbb{W}_c, \ c = c(\mathbf{x}^i) \\ \mathbf{w}^- &:= \mathbf{w}^l : d(x^i, w^l) \leq d(x^i, w^j), \{\mathbf{w}^j, \mathbf{w}^l\} \notin \mathbb{W}_c \end{aligned}$$

Generalized LVQ [26] is derived from a cost function which can be related to the generalization ability of LVQ classifiers [33]:

$$E_{\text{GLVQ}} = \sum_i f\left(\frac{d_i^+ - d_i^-}{d_i^+ + d_i^-}\right) \tag{5}$$

where f is a differentiable monotonic function such as the hyperbolic tangent. Hence, for every data point, its contribution to the cost function is small if and only if the distance to the closest prototype with a correct label is smaller than the distance to a wrongly labeled prototype, resulting in a correct classification of the point and, at the same time, aiming at a large hypothesis margin of the classifier, i.e., a good generalization ability.

A learning algorithm can be derived thereof by means of standard gradient techniques. After presenting data point $\mathbf{x}^i$, its closest correct and wrong prototype, respectively, are adapted according to the prescription:

$$\begin{aligned} \Delta\mathbf{w}^+(\mathbf{x}^i) &\sim -f'(\mu(\mathbf{x}^i)) \cdot \mu^+(\mathbf{x}^i) \cdot \nabla_{\mathbf{w}^+(\mathbf{x}^i)} d_i^+ \\ \Delta\mathbf{w}^-(\mathbf{x}^i) &\sim f'(\mu(\mathbf{x}^i)) \cdot \mu^-(\mathbf{x}^i) \cdot \nabla_{\mathbf{w}^-(\mathbf{x}^i)} d_i^- \end{aligned}$$

where

$$\mu(\mathbf{x}^i) = \frac{d_i^+ - d_i^-}{d_i^+ + d_i^-},$$

$$\mu^+(\mathbf{x}^i) = \frac{2 \cdot d_i^-}{(d_i^+ + d_i^-)^2},$$

$$\mu^-(\mathbf{x}^i) = \frac{2 \cdot d_i^+}{(d_i^+ + d_i^-)^2}.$$

For the squared Euclidean norm, the derivative yields

$$\nabla_{\mathbf{w}^j} d(\mathbf{x}^i, \mathbf{w}^j) = -2(\mathbf{x}^i - \mathbf{w}^j),$$

leading to Hebbian update rules of the prototypes which take into account the priorly known class information.

GLVQ constitutes one particularly efficient method to adapt the prototypes according to a given labeled data sets. Alternatives can be derived based on a labeled Gaussian mixture model, see e.g. [36]. Since the latter can be highly sensitive to model meta-parameters [5], we focus on GLVQ.

4 DISSIMILARITY DATA

Due to improved sensor technology or dedicated data formats, for example, data are becoming more and more complex in many application domains. To account for this fact, data are often addressed by a dedicated dissimilarity measure which respects the structural form of the data such as alignment techniques for bioinformatics sequences, functional norms for mass spectra, or the compression distance for texts [8]. The work in [25] is focused on the theoretical analysis of dissimilarity data and pseudo-euclidean data spaces and motivated our proposed method.

Prototype-based techniques such as GLVQ are restricted to Euclidean vector spaces such that their suitability for complex non-Euclidean data sets is highly limited. Here we propose an extension of GLVQ to general dissimilarity data.

We assume that data $\mathbf{x}^i, i = 1, \ldots, N$ are characterized by pairwise dissimilarities $d_{ij} = d(\mathbf{x}^i, \mathbf{x}^j)$. N denotes the number of data points. D refers to the corresponding dissimilarity matrix in $\mathbb{R}^{N \times N}$. We assume symmetry $d_{ij} = d_{ji}$ and zero diagonal $d_{ii} = 0$. However, D need not correspond to Euclidean data vectors, i.e. it is not guaranteed that data vectors $\mathbf{x}^i$ can be found with $d_{ij} = \|\mathbf{x}^i - \mathbf{x}^j\|^2$.

For every dissimilarity matrix D of this form, an associated similarity matrix is induced by $S = -JDJ/2$ where $J = (I - \mathbf{1}\mathbf{1}^T/N)$ with identity matrix I and vector of ones $\mathbf{1}$. D is Euclidean if and only if S is positive semi-definite (pdf). In general, S displays eigenvectors with p positive eigenvalues, q negative eigenvalues, and $N-p-q$ eigenvalues 0, $(p, q, N-p-q)$ is referred to as the signature.

For kernel methods such as SVM, a correction of the matrix S is necessary to guarantee pdf. Three different techniques are very popular: the spectrum of the matrix S is changed, possible operations being clip (negative eigenvalues are set to 0), flip (absolute values are taken), or shift (a summand is added to all eigenvalues) [8]. Interestingly, some operations such as shift do not affect the location of local optima of important cost functions such as the quantization error [20], albeit the transformation can severely affect the performance of optimization algorithms [16]. As an alternative, data points can be treated as vectors which coefficients are given by the pairwise similarity. These vectors can be processed using standard, e.g. linear or Gaussian kernels. In [8] an extensive comparison of these preprocessing methods in connection to SVM is performed for a variety of benchmarks.

Alternatively, one can directly embed data in the pseudo-Euclidean vector space determined by the eigenvector decomposition of S. Pseudo-Euclidean space is a vector space equipped with a (possible indefinite) symmetric bilinear form which can be used to compute similarities and dissimilarities of data points. More precisely, a symmetric bilinear form is induced by $\langle \mathbf{x}, \mathbf{y} \rangle_{p,q} = \mathbf{x}^T I_{p,q} \mathbf{y}$ where $I_{p,q}$ is a diagonal matrix with p entries 1 and q entries -1. Taking the eigenvectors of S together with the square root of the absolute value of the eigenvalues, we obtain vectors $\mathbf{x}^i$ in pseudo-Euclidean space such that $d_{ij} = \langle \mathbf{x}^i - \mathbf{x}^j, \mathbf{x}^i - \mathbf{x}^j \rangle_{p,q}$ holds for every pair of data points. If the number of data is not limited a priori, a generalization of this concept to Krein spaces which similarly decompose into two possibly infinite dimensional Hilbert spaces is possible [25].

Vector operations can be directly transferred to pseudo-Euclidean space, i.e. we can define prototypes as linear combinations of data in this space. Hence we can perform techniques such as GLVQ explicitly in pseudo-Euclidean space since it relies on vector operations only. One problem of this explicit transfer is given by the computational complexity of the embedding which is $\mathcal{O}(N^3)$, and, further, the fact that out-of-sample extensions to new data points characterized by pairwise dissimilarities are not immediate. Because of this fact, we are interested in efficient techniques which implicitly refer to this embedding only. As a side product, such algorithms are invariant to coordinate transforms in pseudo-Euclidean space.

The key assumption is to restrict prototype positions to linear combinations of data points of the form

$$\mathbf{w}^j = \sum_i \alpha_{ji} \mathbf{x}^i \text{ with } \sum_i \alpha_{ji} = 1 \,.$$

Since prototypes are located at representative points in the data space, it is a reasonable assumption to restrict prototypes to the affine subspace spanned by the given data points. In this case, dissimilarities can be computed implicitly by means of the formula

$$d(\mathbf{x}^i, \mathbf{w}^j) = [D \cdot \alpha_j]_i - \frac{1}{2} \cdot \alpha_j^T D \alpha_j \tag{6}$$

where $\alpha_j = (\alpha_{j1}, \ldots, \alpha_{jn})$ refers to the vector of coefficients describing the prototype $\mathbf{w}^j$ *implicitly*, as shown in [16]. Neither the prototypes nor the original points, related to the dissimilarity matrix, are expected to exist in a vectorial space. This observation constitutes the key to transfer GTM and GLVQ to relational data without an explicit embedding in pseudo-Euclidean space.

5 RELATIONAL GENERATIVE TOPOGRAPHIC MAPPING

GTM has been extended to general dissimilarities in [14]. We shortly recall the approach for convenience. As before, targets $\mathbf{t}^k$ in pseudo-Euclidean space induce a mixture distribution in the data space based on the dissimilarities. Targets are obtained as images of points $\mathbf{w}^k$ in latent space via a generalized linear regression model where, now, the mapping is to the coefficient vectors α which implicitly represent the targets:

$$y : \mathbf{w} \mapsto \boldsymbol{\alpha} = \Phi(\mathbf{w}) \cdot \mathbf{W}$$

with images in $\mathbb{R}^N$ according to the dimensionality of the coefficients α.

The restriction

$$\sum_i [\Phi(\mathbf{w}k) \cdot \mathbf{W}]_i = \sum_i \alpha_{ki} = 1$$

is automatically fulfilled for optima of the data log likelihood. Hence the likelihood function can be computed based on (1) and the distance computation can be performed indirectly using (6). An EM optimization scheme leads to solutions for the parameters β and $\mathbf{W}$, and an expression for the hidden variables given by the responsibilities of the modes for the data points. Algorithmically, Eqn. (2) using (6) and the optimization of the expectation

$$\sum_{k,i} R_{ki}(\mathbf{W}_{\text{old}}, \beta_{\text{old}}) \ln p(\mathbf{x}^i | \mathbf{w}k, \mathbf{W}_{\text{new}}, \beta_{\text{new}})$$

with respect to $\mathbf{W}$ and β take place in turn. The latter yields model parameters which can be determined in analogy to (3,4) where, now, functions Φ map from the latent space to the space of coefficients α and $\mathbf{X}$ denotes the unity matrix in the space of coefficients. We refer to this iterative update scheme as relational GTM (RGTM). Initialization takes place by referring to the first MDS directions of $\mathbf{D}$. See [14] for details.

6 RELATIONAL LEARNING VECTOR QUANTIZATION

We use the same principle to extend GLVQ to relational data. Again, we assume a symmetric dissimilarity matrix D with zero diagonal is given. We assume that a prototype $\mathbf{w}^j$ is represented implicitly by means of the coefficient vectors α_j. Then we can use the equivalent characterization of distances (6) in the GLVQ cost function (5) leading to the costs of relational GLVQ (RGLVQ):

$$\begin{aligned} E_{\mathrm{RGLVQ}} &= \sum_i f\left(\frac{\xi(i)^+ - \zeta(i)^+ - \xi(i)^- + \zeta(i)^-}{\xi(i)^+ - \zeta(i)^+ + \xi(i)^- - \zeta(i)^-}\right) \\ \xi(i)^+ &= [D\alpha^+]_i \\ \xi(i)^- &= [D\alpha^-]_i \\ \zeta(i)^+ &= \frac{1}{2}\cdot(\alpha^+)^T D\alpha^+ \\ \zeta(i)^- &= \frac{1}{2}\cdot(\alpha^-)^T D\alpha^- \end{aligned}$$

where as before the closest correct and wrong prototype are referred to, corresponding to the coefficients α^+ and $\alpha-$, respectively. A simple stochastic gradient descent leads to adaptation rules for the coefficients α^+ and α^- in relational GLVQ: component k of these vectors is adapted as

$$\begin{aligned} \Delta\alpha_k^+ &\sim \frac{-\Phi'(\mu(\mathbf{x}^i))}{(\mu^+(\mathbf{x}^i))^{-1}} \cdot \frac{\partial\left([D\alpha^+]_i - \frac{1}{2}(\alpha^+)^T D\alpha^+\right)}{\partial\alpha_k^+} \\ \Delta\alpha_k^- &\sim \frac{\Phi'(\mu(\mathbf{x}^i))}{(\mu^-(\mathbf{x}^i))^{-1}} \cdot \frac{\partial\left([D\alpha^-]_i - \frac{1}{2}(\alpha^-)^T D\alpha^-\right)}{\partial\alpha_k^-} \end{aligned}$$

where $\mu(\mathbf{x}^i)$, $\mu^+(\mathbf{x}^i)$, and $\mu^-(\mathbf{x}^i)$ are as above. The partial derivative yields

$$\frac{\partial\left([D\alpha_j]_i - \frac{1}{2}\cdot\alpha_j^T D\alpha_j\right)}{\partial\alpha_{jk}} = d_{ik} - \sum_l d_{lk}\alpha_{jl}$$

Naturally, alternative gradient techniques such as line search can be used in a similar way.

After every adaptation step, normalization takes place to guarantee $\sum_i \alpha_{ji} = 1$. This way, a learning algorithm which adapts prototypes in a supervised manner similar to GLVQ is given for general dissimilarity data, whereby prototypes are implicitly embedded in pseudo-Euclidean space. The prototypes are initialized as random vectors, i.e we initialize α_{ij} with small random values such that the sum is one. It is possible to take class information into account by setting all α_{ij} to zero which do not correspond to the class of the prototype.

For both, RGTM and RGLVQ, out-of-sample extension of the model to new data is possible immediately. It can be based on an observation made in [16]: given a novel data point $\mathbf{x}$ characterized by its pairwise dissimilarities $D(\mathbf{x})$ to the data vectors $\mathbf{x}^i$ used for training, the dissimilarity of $\mathbf{x}$ to a prototype represented by α_j is

$$d(\mathbf{x}, \mathbf{w}^j) = D(\mathbf{x})^T \cdot \alpha_j - \frac{1}{2}\cdot\alpha_j^T D\alpha_j.$$

This can be directly used to compute responsibilities for RGTM or the closest prototype for RGLVQ, respectively.

7 THE NYSTRÖM APPROXIMATION

Both techniques, RGTM and RGLVQ depend on the full dissimilarity matrix D. This is of size N^2, hence the techniques have quadratic complexity with respect to the given number of data points. This is infeasible for large N: restrictions are given by the main memory (assuming double precision and 12 GB main memory, the limit is currently at about 30,000 data points), and the time necessary to compute dissimilarities and train the models based thereon (assuming 1ms for one dissimilarity computation, which is quite reasonable for complex dissimilarities e.g. based on alignment techniques, a matrix of less than 10,000 data points can be computed in 12 h on a dual core machine.) Therefore, approximation techniques which reduce the effort to a linear one would be very desirable.

7.1 Nyström approximation for similarity data

Nyström approximation technique has been proposed in the context of kernel methods in [38]. Here, we give a short review of this technique.

One well known way to approximate a $N \times N$ Gram matrix, is to use a low-rank approximation. This can be done by computing the eigendecomposition of the kernel $\mathbf{K} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{U}$ is a matrix, whose columns are orthonormal eigenvectors, and $\mathbf{\Lambda}$ is a diagonal matrix consisting of eigenvalues $\mathbf{\Lambda}_{11} \geq \mathbf{\Lambda}_{22} \geq ... \geq 0$, and keeping only the m eigenspaces which correspond to the m largest eigenvalues of the matrix. The approximation is $\mathbf{K} \approx \mathbf{U}_{N,m}\mathbf{\Lambda}_{m,m}\mathbf{U}_{m,N}$, where the indices refer to the size of the corresponding submatrix. The Nyström method approximates a kernel in a similar way, without computing the eigendecomposition of the whole matrix, which is an $O(N^3)$ operation.

By the Mercer theorem kernels $k(\mathbf{x}, \mathbf{y})$ can be expanded by orthonormal eigenfunctions ψ_i and non negative eigenvalues λ_i in the form

$$k(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{\infty} \lambda_i \psi_i(\mathbf{x}) \psi_i(\mathbf{y}).$$

The eigenfunctions and eigenvalues of a kernel are defined as the solution of the integral equation

$$\int k(\mathbf{y}, \mathbf{x}) \psi_i(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = \lambda_i \psi_i(\mathbf{y}),$$

where $p(\mathbf{x})$ is the probability density of $\mathbf{x}$. This integral can be approximated based on the Nyström technique by sampling $\mathbf{x}^k$ i.i.d. according to $p(\mathbf{x})$:

$$\frac{1}{m} \sum_{k=1}^{m} k(\mathbf{y}, \mathbf{x}^k) \psi_i(\mathbf{x}^k) \approx \lambda_i \psi_i(\mathbf{y}).$$

Using this approximation and the matrix eigenproblem equation

$$\mathbf{K}^{(m)}\mathbf{U}^{(m)} = \mathbf{U}^{(m)}\mathbf{\Lambda}^{(m)}$$

of the corresponding $m \times m$ Gram sub-matrix $\mathbf{K}^{(m)}$ we can derive the approximations for the eigenfunctions and eigenvalues of the kernel k

$$\lambda_i \approx \frac{\lambda_i^{(m)}}{m}, \quad \psi_i(\mathbf{y}) \approx \frac{\sqrt{m}}{\lambda_i^{(m)}} \mathbf{k}_y \mathbf{u}_i^{(m)}, \tag{7}$$

where $\mathbf{u}_i^{(m)}$ is the ith column of $\mathbf{U}^{(m)}$. Thus, we can approximate ψ_i at an arbitrary point $\mathbf{y}$ as long as we know the vector $\mathbf{k}_y = (k(\mathbf{x}^1, \mathbf{y}), ..., k(\mathbf{x}^m, \mathbf{y}))^T$.

For a given $N \times N$ Gram matrix $\mathbf{K}$ we randomly choose m rows and respective columns. The corresponding indices are also called landmarks, and should be chosen such that the data distribution is sufficiently covered. A specific analysis about selection strategies was recently discussed in [40]. We denote these rows by $\mathbf{K}_{m,N}$. Using the formulas (7) we obtain $\tilde{\mathbf{K}} = \sum_{i=1}^{m} 1/\lambda_i^{(m)} \cdot \mathbf{K}_{m,N}^T \mathbf{u}_i^{(m)} (\mathbf{u}_i^{(m)})^T \mathbf{K}_{m,N}$, where $\lambda_i^{(m)}$ and $\mathbf{u}_i^{(m)}$ correspond to the $m \times m$ eigenproblem. Thus we get, $\mathbf{K}_{m,m}^{-1}$ denoting the Moore-Penrose pseudoinverse,

$$\tilde{\mathbf{K}} = \mathbf{K}_{m,N}^t \mathbf{K}_{m,m}^{-1} \mathbf{K}_{m,N}. \tag{8}$$

as an approximation of $\mathbf{K}$. This approximation is exact, if $\mathbf{K}_{m,m}$ has the same rank as $\mathbf{K}$.

7.2 Nyström approximation for dissimilarity data

For dissimilarity data, a direct transfer is possible, see [13] for preliminary work on this topic. According to the spectral theorem, a symmetric dissimilarity matrix $\mathbf{D}$ can be diagonalized $\mathbf{D} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ with $\mathbf{U}$ being a unitary matrix whose column vectors are the orthonormal eigenvectors of $\mathbf{D}$ and $\mathbf{\Lambda}$ a diagonal matrix with the eigenvalues of $\mathbf{D}$, which can be negative for non-Euclidean distances. Therefore the dissimilarity matrix can be seen as an operator

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} \lambda_i \psi_i(\mathbf{x}) \psi_i(\mathbf{y})$$

where $\lambda_i \in \mathbb{R}$ correspond to the diagonal elements of $\mathbf{\Lambda}$ and ψ_i denote the eigenfunctions. The only difference to an expansion of a kernel is that the eigenvalues can be negative. All further mathematical manipulations can be applied in the same way.

Using the approximation (8) for the distance matrix, we can apply this result for RGTM. It allows to approximate dissimilarities between a prototype $\mathbf{w}^k$ represented by a coefficient vector α_k and a data point $\mathbf{x}^i$ in the way

$$\begin{aligned} d(\mathbf{x}^i, \mathbf{w}^k) \quad \approx \quad & \left[\mathbf{D}_{m,N}^T \left(\mathbf{D}_{m,m}^{-1} \left(\mathbf{D}_{m,N} \boldsymbol{\alpha}_k\right)\right)\right]_i \\ & -\frac{1}{2} \cdot \left(\boldsymbol{\alpha}_k^T \mathbf{D}_{m,N}^T\right) \cdot \\ & \left(\mathbf{D}_{m,m}^{-1} \left(\mathbf{D}_{m,N} \boldsymbol{\alpha}_k\right)\right) \end{aligned} \tag{9}$$

with a linear submatrix of m rows and a low rank matrix $\mathbf{D}_{m,m}$. Performing these matrix multiplications from right to left, this computation is $\mathcal{O}(m^2 N)$ instead of $\mathcal{O}(N^2)$, i.e. it is linear in the number of data points N, assuming fixed approximation m.

We use this approximation directly in RGTM and RGLVQ by taking a random subsample of m points to approximate the dissimilarity matrix. The percentage m is differed during training showing the effect of the approximation on the percentage used for the approximation.

A benefit of the Nyström technique is that it can be decided priorly which linear parts of the dissimilarity matrix will be used in training. Therefore, it is sufficient to compute only a linear part of the full dissimilarity matrix D to use these methods. A drawback of the Nyström approximation is that a good approximation can only be achieved if the rank of $\mathbf{D}$ is kept as much as possible, i.e. the chosen subset should be representative. We will see that the method can be used in many (though not all) settings leading to a considerable speed-up.

8 EXPERIMENTS

We evaluate the techniques on three benchmarks from the biomedical domain:

- The *Copenhagen Chromosomes data set* constitutes a benchmark from cytogenetics [22]. A set of 4,200 human chromosomes from 21 classes (the autosomal chromosomes) are represented by grey-valued images. These are transferred to strings measuring the thickness of their silhouettes. These strings are compared using edit distance with insertion/deletion costs 4.5.

- The *vibrio data set* consists of 1,100 samples of vibrio bacteria populations characterized by mass spectra. The spectra encounter approx. 42,000 mass positions. The full data set consists of 49 classes of vibrio-sub-species. The mass spectra are preprocessed with a standard workflow using the BioTyper software [23]. According to the functional form of mass spectra, dedicated similarities as provided by the BioTyper software are used [23].

- Similar to an application presented in [19], we extract roughly 11,000 protein sequences of the *SwissProt data base* according to 32 functional labels given by PROSITE [11]. Sequence alignment is done using local alignment by means of the Smith-Waterman algorithm.

We compare the two different prototype-based methods, RGTM and RGLVQ and the Nyström approximations thereof. The following parameters are used:

- Evaluation is done by means of the generalization error in a ten fold repeated cross-validation with ten repeats. (Only two-fold cross-validation and five repeats for SwissProt.) For RGTM, posterior labeling on the test set is used.

- The number of prototypes is chosen roughly following the number of priorly known classes. For RGLVQ, we use 63 prototypes for the Chromosomes data set, 49 prototypes for the Vibrio data set, and 64 prototypes for the SwissProt data set. For RGTM, to account for the topological constraints which result in prototypes outside the convex hull of the data, we use lattices of size 20×20 for Chomosomes and Vibrio and 40×40 for SwissProt. 10×10 base functions are used in both cases.

- Training takes place until convergence which is 50 epochs for the small data sets and 5 epochs for SwissProt.

- For the Nyström approximation, we report the results obtained when sampling 1% and 10% of the data.

- For the SwissProt data set, the speed up of the method can clearly be observed due to the large size of the data set. We also report the CPU time in seconds taken for one cross-validation run on a 24 Intel(R) Xeon X5690 machine with 3.47GHz processors and 48 GB DDR3 1333MHz memory. The experiments are implemented in Matlab.

The results of the techniques are collected in Tab. 1. The transformations for the SVM are done in accordance to [8] with results taken form [27] [2]. Since there is not yet a best way to do this eigenvalue correction, all approaches have to be tried on the training data and the best results is chosen. Most of these transformation require a singular value decomposition of the similarity matrix, with a complexity of $(O(N^3))$. In contrast the proposed relational methods do not require any kind of preprocessing but can be applied directly on the given, symmetric, dissimilarity matrices.

For both data sets, Chromosomes and Vibrio, the classification accuracy of RGLVQ is better as compared to the unsupervised RGTM which can be attributed to the fact that the primal can take the priorly known class information into account during training. Interestingly, the results of the Nyström approximation are quite diverse. In some cases, the

[2] SVM results are obtained by a standard C++ implementation, while the other experiments are done in pure matlab, hence the CPU time is not comparable here.

	Chromosome	*Vibrio*	*SwissProt*	CPU
RGTM	88.10 (0.7)	94.70 (0.5)	69.90 (2.5)	9656
RGTM (Ny 0.01)	87.80 (2.70)	54.70 (3.10)	74.40 (2.70)	786
RGTM (Ny 0.10)	51.60 (6.60)	93.90 (0.70)	82.20 (4.50)	1631
RGLVQ	92.70 (0.20)	100.00(0.00)	82.30(0.00)	24481
RGLVQ (Ny 0.01)	78.40 (0.10)	99.10(0.10)	87.00(0.00)	4179
RGLVQ (Ny 0.10)	78.20 (0.40)	99.20(0.20)	83.40(0.20)	9696
SVM*	92.50 (3.30)	100.00(0.00)	98.40(0.10)	-
SVM* **(Ny 0.01)**	95.60 (1.30)	85.27(4.32)	86.30(0.10)	-
SVM* **(Ny 0.1)**	68.80 (1.90)	99.82(0.57)	63.00(1.50)	-

Table 1: Results of the methods on the three data sets, the generalization error is reported, the standard deviation is given in parentheses. For SwissProt, we also report the CPU time for one run in seconds.

classification accuracy is nearly the same as for the original method, in others, the accuracy even increases when taking the Nyström approximation. In some cases, the result drops down by more than 40%. Interestingly, the result is not monotonic with respect to the size used to approximate the data, and it is also not consistent for the two algorithms. While Nyström approximation is clearly possible for RGLVQ and the Vibrio data set, the quality depends very much on the approximation parameters for RGTM.

Thus it seems that the quality of the approximation is not necessarily better the larger the fraction of the data taken for approximation, and it seems that the techniques are affected to a different degree by this approximation quality.

To shed some light on this aspect, we directly evaluate the quality of the Nyström approximation as follows: we repeatedly sample a different fraction of the data set and evaluate the distance of the approximated matrix and the original one. Since both methods do not depend on the exact size of the dissimilarities, but rather the ranking induced by the values is important, we evaluate the spearman correlation of the resulting columns. The results are depicted in Fig. 1,2. Interestingly, the resulting quality is not monotonic with respect to the size of the subsample taken for the approximation. Rather, the spearman correlation drops down for all settings and larger percentage of the subsample for all three cases. This can probably be attributed to the fact that, for larger values, noise in the data accounts for random fluctuations of the ranks rather than an approximation of the underlying order. Hence it can be advisable to test different, on particular also comparably small subsamples to arrive at a good approximation.

The speed-up of the techniques by means of the approximation has been evaluated for the SwissProt data set as a comparably large data set. Note that the current limit regarding memory restrictions for a standard memory size of 12 GB would allow at most 30,000 samples, hence the SwissProt data data also in the order of magnitude of this limit. Interestingly, the speed-up is more than 2.5 if 10% are taken and close to six if only 1% is chosen for RGLVQ. Hence the Nyström approximation can contribute to a considerable speed-up in these cases, while not deteriorating the quality for RGLVQ or RGTM.

9 CONCLUSIONS

Relational GTM offers a highly flexible tool to simultaneously cluster and order dissimilarity data in a topographic mapping. It relies on an implicit pseudo-Euclidean embedding of data such that dissimilarities become directly accessible. We have proposed a similar extension

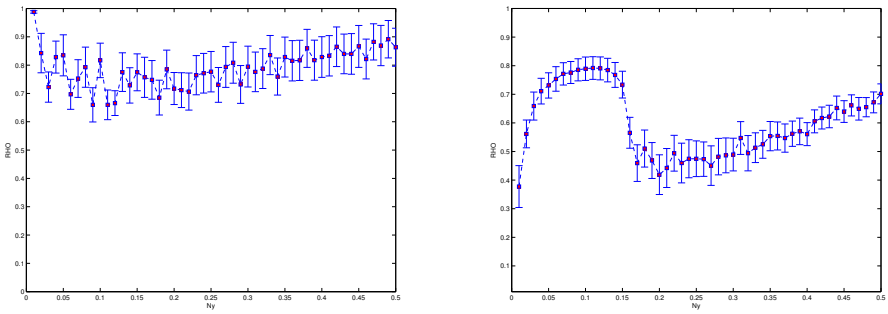


Figure 1: Quality of the Nyström approximation as evaluated by the Spearman correlation of the rows of the approximated matrix and the original one. The approximation is based on a different fraction of the data set as indicated by the x-axis. The graphs show the result for the Chromosomes (left) and Vibrio (right).

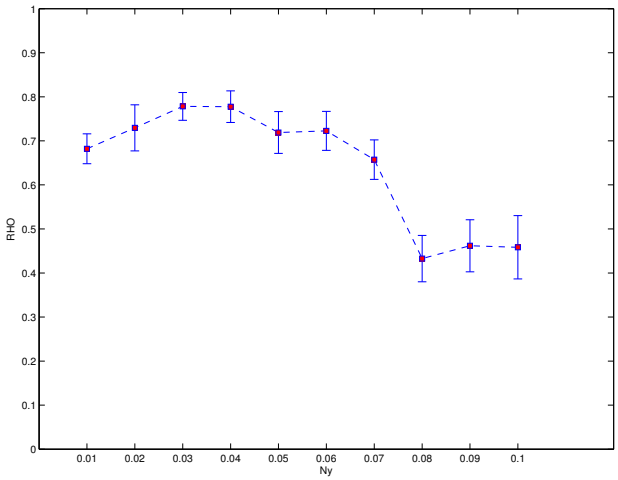


Figure 2: Quality of the Nyström approximation as evaluated by the Spearman correlation for SwissProt.

of supervised prototype based methods, more precisely GLVQ, to obtain a high quality classification scheme for dissimilarity data.

Due to the dependency on the full matrix, both methods requires squared time complexity and memory to store the dissimilarities. We have proposed a speed-up techniques which leads to linear effort: the Nyström approximation. Using three examples from the biomedical domain, we demonstrated that already for comparably small data sets the technique can largely enhance speed while not loosing too much information contained in the data.

Interestingly, the quality of the Nyström technique does not scale monotonously with the sample size taken for the approximation. Rather, depending on the data characteristics, smaller samples might lead to a better job. Therefore, it is always worthwhile to test different sample sizes to achieve the optimum balance of accuracy and speed.

Acknowledgment

This work was supported by the "German Science Foundation (DFG)" under grant number HA-2719/4-1. Further, financial support from the Cluster of Excellence 277 Cognitive Interaction Technology funded in the framework of the German Excellence Initiative is gratefully acknowledged. We would like to thank Dr. Markus Kostrzewa, Dr. Thomas Maier, Dr. Stephan Klebel, Dr. Thomas Elssner and Dr. Karl-Otto Kruter (all Bruker Daltonik GmbH), for providing the Vibrio data set and additional support during pre-processing and interpretation.

References

[1] N. Alex, A. Hasenfuss, and B. Hammer. Patch clustering for massive data sets. *Neurocomputing*, 72(7-9):1455–1469, 2009.

[2] W. Arlt, M. Biehl, A.E. Taylor, S. Hahner, R. Libe, B.A. Hughes, P. Schneider, D.J. Smith, H. Stiekema, N. Krone, E. Porfiri, G. Opocher, J. Bertherat, F. Mantero, B. Allolio, M. Terzolo, P. Nightingale, C.H.L. Shackleton, X. Bertagna, M. Fassnacht, P.M. Stewart Urine Steroid Metabolomics as a Biomarker Tool for Detecting Malignancy in Adrenal Tumors *J. of Clinical Endocrinology & Metabolism*, Vol. 96, No. 12, pp. 3775-3784, 2011

[3] Wesam Barbakh and Colin Fyfe. Online clustering algorithms. *Int. J. Neural Syst.*, 18(3):185–194, 2008.

[4] S. B. Barbuddhe, T. Maier, G. Schwarz, M. Kostrzewa, H. Hof, E. Domann, T. Chakraborty, and T. Hain, Rapid identification and typing of listeria species by matrix-assisted laser desorption ionization-time of flight mass spectrometry, *Applied and Environmental Microbiology*, vol. 74, no. 17, pp. 5402–5407, 2008.

[5] M. Biehl, A. Ghosh, and B. Hammer, Dynamics and generalization ability of LVQ algorithms, J. Machine Learning Research 8 (Feb):323-360, 2007.

[6] C. Bishop, M. Svensen, and C. Williams. The generative topographic mapping. *Neural Computation* 10(1):215-234, 1998.

[7] Romain Boulet, Bertrand Jouve, Fabrice Rossi and Nathalie Villa-Vialaneix. Batch kernel SOM and related Laplacian methods for social network analysis. *Neurocomputing*, 71(7-9:1257-1273, 2008.

[8] Y. Chen, E. K. Garcia, M. R. Gupta, A. Rahimi, L. Cazzanti; Similarity-based Classification: Concepts and Algorithms, Journal of Machine Learning Research 10(Mar):747–776, 2009.

[9] J. Shawe-Taylor and N. Cristianini, Kernel Methods for Pattern Analysis and Discovery *Cambridge University Press*, 2004

[10] M. Cottrell, B. Hammer, A. Hasenfuss, and T. Villmann. Batch and median neural gas. *Neural Networks*, 19:762–771, 2006.

[11] E. Gasteiger, A. Gattiker, C. Hoogland, I. Ivanyi, R.D. Appel, A. Bairoch, ExPASy: the proteomics server for in-depth protein knowledge and analysis, Nucleic Acids Res. 31:3784-3788, 2003.

[12] Roberto Gil-Pita and Xin Yao. Evolving edited k-nearest neighbor classifiers. *Int. J. Neural Syst.*, 18(6):459–467, 2008.

[13] A. Gisbrecht, B. Mokbel, and B. Hammer. The Nystrom approximation for relational generative topographic mappings. In *NIPS workshop on challenges of Data Visualization*, 2010.

[14] A. Gisbrecht, B. Mokbel, and B. Hammer. Relational Generative Topographic Mapping. Neurocomputing 74: 1359-1371, 2011.

[15] T. Graepel and K. Obermayer (1999), A stochastic self-organizing map for proximity data, *Neural Computation* 11:139-155, 1999.

[16] B. Hammer and A. Hasenfuss. Topographic Mapping of Large Dissimilarity Data Sets. Neural Computation 22(9):2229-2284, 2010.

[17] R. J. Hathaway and J. C. Bezdek. Nerf c-means: Non-Euclidean relational fuzzy clustering. *Pattern Recognition* 27(3):429-437, 1994.

[18] T. Kohonen, editor. *Self-Organizing Maps*. Springer-Verlag New York, Inc., 3rd edition, 2001.

[19] T. Kohonen and P. Somervuo (2002), How to make large self-organizing maps for non-vectorial data, *Neural Networks* 15:945-952.

[20] J. Laub, V. Roth, J.M. Buhmann, K.-R. Müller. On the information and representation of non-Euclidean pairwise data. *Pattern Recognition* 39:1815-1826 2006.

[21] Ezequiel López-Rubio, Rafael Marcos Luque Baena, and Enrique Domínguez. Foreground detection in video sequences with probabilistic self-organizing maps. *Int. J. Neural Syst.*, 21(3):225–246, 2011.

[22] C. Lundsteen, J. Phillip, and E. Granum (1980), Quantitative analysis of 6985 digitized trypsin G-banded human metaphase chromosomes, *Clinical Genetics* 18:355-370.

[23] T. Maier, S. Klebel, U. Renner, and M. Kostrzewa, Fast and reliable MALDI-TOF ms–based microorganism identification, *Nature Methods*, no. 3, 2006.

[24] Britta Mersch, Tobias Glasmachers, Peter Meinicke, and Christian Igel. Evolutionary optimization of sequence kernels for detection of bacterial gene starts. *Int. J. Neural Syst.*, 17(5):369–381, 2007.

[25] E. Pekalska and R.P.W. Duin The Dissimilarity Representation for Pattern Recognition. Foundations and Applications. World Scientific, Singapore, December 2005.

[26] A. Sato and K. Yamada. Generalized learning vector quantization. In M. C. Mozer D. S. Touretzky and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8. Proceedings of the 1995 Conference*, pages 423–9, Cambridge, MA, USA, 1996. MIT Press.

[27] B. Hammer and B. Mokbel and F.-M. Schleif and X. Zhu White Box Classification of Dissimilarity Data. In E. Corchado and V. Snásel and A. Abraham and M. Wozniak and M. Graña and S.-B. Cho, editors, *Hybrid Artificial Intelligent Systems - 7th International Conference, HAIS 2012, Salamanca, Spain, March 28-30th, 2012. Proceedings, Part I*, pages 309-321, Springer.

[28] B. Hammer and F.-M. Schleif and X. Zhu Relational Extensions of Learning Vector Quantization. In B.-L. Lu and L. Zhang and J. T. Kwok, editors, *Neural Information Processing - 18th International Conference, ICONIP 2011, Shanghai, China, November 13-17, 2011, Proceedings, Part II*, pages 481-489, Springer.

[29] Efficient kernelized prototype based classification, Frank-M. Schleif, T. Villmann, B. Hammer, P. Schneider, M. Biehl, International Journal of Neural Systems, vol. 21, no, 6, pp. 443-457, 2011.

[30] Ranking-based Kernels in Applied Biomedical Diagnostics using Support Vector Machine, V. Jumutc, P. Zayakin, A. Borisov. International Journal of Neural Systems, vol. 21, no, 6, pp. 459-473, 2011.

[31] Discovering Significant Evolution Patterns from Satellite Image Time Series, F. Petitjean, F. Masseglia, P. Gancarski, , and G. Forestier International Journal of Neural Systems, vol. 21, no, 6, pp. 475-489, 2011.

[32] E. Pekalska, R. P. W. Duin, S. Günter and H. Bunke On Not Making Dissimilarities Euclidean *SSPR/SPR'2004*, pp. 1145-1154, 2004

[33] P. Schneider, M. Biehl, and B. Hammer, Adaptive relevance matrices in learning vector quantization,' *Neural Computation*, vol. 21, no. 12, pp. 3532–3561, 2009.

[34] Principal Manifolds and Graphs in Practice: From Molecular Biology to Dynamical Systems, A.N. Gorban and A. Zinovyev International Journal of Neural Systems, vol. 20, no, 3, pp. 219-232, 2011.

[35] S. Seo and K. Obermayer (2004), Self-organizing maps and clustering methods for matrix data, *Neural Networks* 17:1211-1230.

[36] S. Seo and K. Obermayer. Soft learning vector quantization. *Neural Computation*, 15(7):1589–1604, 2003.

[37] L. Shi and Y. Shi and Y. Gao A Novel Approach of Clustering with an Extended Classifier System *International Journal of Neural Systems*, 21(1):79–93, 2011.

[38] C. K. I. Williams, M. Seeger. Using the Nyström method to speed up kernel machines. *Advances in Neural Information Processing Systems 13*: 682-688, 2001

[39] H. Yin. On the equivalence between kernel self-organising maps and self-organising mixture density networks. *Neural Networks*, 19(6-7):780–784, 2006.

[40] K. Zhang and J. T. Kwok. Clustered Nyström method for large scale manifold learning and dimension reduction *IEEE Transactions on Neural Networks*, 21(10): 1576-1587, 2010

Chapter 5

Acknowledgment

The research presented in this thesis was done over multiple year at different locations and in various research environments. I met many interesting and inspiring people who made all this possible and have been supportive in different ways. Early stones about *fuzzy*-labeling methods, not used in my PhD thesis, date back to work for Bruker Biosciences. There, Dr. Jens Decker the team leader of the *numerical toolbox*-group was brave enough to bring in a young and fresh computer scientist into a team of mathematicians, chemists and engineers to show him, not only the world of real software development in a larger company, but also to let freedom for own research. This early still lasting collaboration motivated a lot of my problem oriented work. I was also a guest in the group of Dr. Markus Kostrzewa, Director of Bioanalytics at Bruker who came and still comes up with a lot of challenging data analysis questions. I would like to thank him and his group for the many interesting data sets and classification and clustering problems he raised. Further, I am supported from the academic field by multiple groups I am cooperating with since multiple year or have been part of. First of all Prof. Thomas Villmann and the Computational Intelligence Group in Mittweida. He made many things possible and was an ongoing and outstanding inspiration for multiple research questions and also social activities to overcome some dry spells during the years. I also would like to thank the group of Prof. Michael Biehl in Groningen, NL which I had the pleasure to join multiple times at longer research stays within research cooperations and different Erasmus-Teaching programs. Luckily, I had many very good master (Stephan Simmuteit, Jessica Brinkmann, Sven Wiesenmüller, Raphael Reisch) and PhD students (Matthias Ongyerth, Tina Geweniger, Dietlind Zühlke, Bassam Mokbel, Andrej Gisbrecht, Xibin Zhu) on the way – thanks. I also would like to thank my many co-authors from very different disciplines. Also The *Natural Computation Group* in Birmingham, especially Reader Peter Tino and Lecturer Ata Kaban have been very inspiring in the last years of this habilitation and opened my mind to many alternative views in computational intelligence. I also have to thank the many funding organization having supported my research during the years: German Research Foundation, the Federal Ministry of Education and Research and the German Academic Exchange Service. The last years I enjoyed research in the group of Theoretical Computer Science in Bielefeld, led by Barbara who was and is a invaluable mentor over the years, thanks a lot. Here, I also want to say that I enjoyed the team spirit and effective collaborative work in the group of Theoretical Computer Science, thanks to all of you. Of course, my warmest thanks go to my parents and my girl-friend Ute, for the many extra hours and days I spend on research, instead of enjoying life with them. Finally, I would like to express my honest thanks to the kind persons who will be willing to review this Habilitation and the person in charge for the final report about my work.

Bibliography

[1] N. Aghaeepour, G. Finak, The FlowCAP Consortium, The DREAM Consortium, H. Hoos, T.R. Mosmann, R. Brinkman, R. Gottardo, and R.H. Scheuermann. Critical assessment of automated flow cytometry data analysis techniques. *Nature Methods*, 10(3):228–238, 2013.

[2] N. Alex, A. Hasenfuss, and B. Hammer. Patch clustering for massive data sets. *Neurocomputing*, 72(7-9):1455–1469, 2009.

[3] J.M. Alonso and L. Magdalena. Special issue on interpretable fuzzy systems. *Information Sciences*, 181(20):4331–4339, 2011.

[4] W. Arlt, M. Biehl, A.E. Taylor, S. Hahner, R. Lib, B.A. Hughes, P. Schneider, D.J. Smith, H. Stiekema, N. Krone, E. Porfiri, G. Opocher, J. Bertherat, F. Mantero, B. Allolio, M. Terzolo, P. Nightingale, C.H.L. Shackleton, X. Bertagna, M. Fassnacht, and P.M. Stewart. Urine steroid metabolomics as a biomarker tool for detecting malignancy in adrenal tumors. *Journal of Clinical Endocrinology and Metabolism*, 96(12):3775–3784, 2011.

[5] B. Arnonkijpanich, A. Hasenfuss, and B. Hammer. Local matrix adaptation in topographic neural maps. *Neurocomputing*, 74(4):522–539, 2011.

[6] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

[7] Mihai Badoiu, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. In *STOC*, pages 250–257, 2002.

[8] V. Balachandran, Deepak P, and D. Khemani. Interpretable and reconfigurable clustering of document datasets by deriving word-based rules. *Knowledge and Information Systems*, 32(3):475–503, 2012.

[9] J. Barnes and P. Hut. A hierarchical o(n log n) force-calculation algorithm. *Nature*, 324(6096):446–449, 1986.

[10] Hans-Ulrich Bauer, J. Michael Herrmann, and Thomas Villmann. Neural maps and topographic vector quantization. *Neural Networks*, 12(4-5):659–676, 1999.

[11] Vanya Van Belle and Paulo J. G. Lisboa. Research directions in interpretable machine learning models. In *ESANN*, 2013.

[12] Amit Bermanis, Amir Averbuch, and Ronald R. Coifman. Multiscale data sampling and function extension. *Appl. Comput. Harmon. Anal.*, 34(1):15–29, 2013.

[13] M. Biehl, A. Ghosh, and B. Hammer. Dynamics and generalization ability of lvq algorithms. *Journal of Machine Learning Research*, 8:323–360, 2007.

[14] M. Biehl, B. Hammer, P. Schneider, and T. Villmann. *Metric learning for prototype-based classification*, volume 247 of *Studies in Computational Intelligence*. 2009.

[15] M. Biehl, B. Hammer, P. Schneider, and T. Villmann. Metric learning for prototype-based classification. *Studies in Computational Intelligence*, 247:183–199, 2009.

[16] Michael Biehl, Kerstin Bunte, Frank-Michael Schleif, Petra Schneider, and Thomas Villmann. Large margin linear discriminative visualization by matrix relevance learning. In He [61], pages 1–8.

[17] Michael Biehl, Anarta Ghosh, and Barbara Hammer. Dynamics and generalization ability of lvq algorithms. *Journal of Machine Learning Research*, 8:323–360, 2007.

[18] Michael Biehl, Barbara Hammer, Erzsébet Merényi, Alessandro Sperduti, and Thomas Villman. Learning in the context of very high dimensional data (Dagstuhl Seminar 11341). *Dagstuhl Reports*, 1(8):67–95, 2011.

[19] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2007.

[20] Christopher M. Bishop, Markus Svensén, and Christopher K. I. Williams. Gtm: The generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998.

[21] T. Bojer, B. Hammer, D. Schunk, and K.T. Von Toschanowitz. Relevance determination in learning vector quantization. *Proceedings of the European Symposium on Artificial Neural Networks (ESANN 2001)*, pages 271–276, 2001.

[22] Edwin V. Bonilla and Antonio Robles-Kelly. Discriminative probabilistic prototype learning. In *ICML*. icml.cc / Omnipress, 2012.

[23] E.V. Bonilla and A. Robles-Kelly. Discriminative probabilistic prototype learning. volume 1, pages 735–742, 2012.

[24] R. Boulet, B. Jouve, F. Rossi, and N. Villa. Batch kernel som and related laplacian methods for social network analysis. *Neurocomputing*, 71(7-9):1257–1273, 2008.

[25] Sebastian Briesemeister, Jörg Rahnenführer, and Oliver Kohlbacher. Going from where to why - interpretable prediction of protein subcellular localization. *Bioinformatics*, 26(9):1232–1238, 2010.

[26] K. Bunte, B. Hammer, A. Wismüller, and M. Biehl. Adaptive local dissimilarity measures for discriminative dimension reduction of labeled data. *Neurocomputing*, 73(7-9):1074–1092, 2010.

[27] K. Bunte, P. Schneider, B. Hammer, F.-M. Schleif, T. Villmann, and M. Biehl. Limited rank matrix learning discriminative dimension reduction and visualization. *Neural Networks*, 26:159–173, 2012.

[28] Kerstin Bunte, Michael Biehl, and Barbara Hammer. A general framework for dimensionality-reducing data visualization mapping. *Neural Computation*, 24(3):771–804, 2012.

[29] Kerstin Bunte, Sven Haase, Michael Biehl, and Thomas Villmann. Stochastic neighbor embedding (sne) for dimension reduction and visualization using arbitrary divergences. *Neurocomputing*, 90(0):23 – 45, 2012. ¡ce:title¿Advances in artificial neural networks, machine learning, and computational intelligence (ESANN 2011)¡/ce:title¿.

[30] Kerstin Bunte, Barbara Hammer, Thomas Villmann, Michael Biehl, and Axel Wismüller. Neighbor embedding xom for dimension reduction and visualization. *Neurocomputing*, 74(9):1340 – 1350, 2011. ¡ce:title¿Advances in artificial neural networks, machine learning, and computational intelligence -¡/ce:title¿ ¡ce:subtitle¿Selected papers from the 18th European Symposium on Artificial Neural Networks (ESANN 2010)¡/ce:subtitle¿.

[31] Kerstin Bunte, Frank-Michael Schleif, and Michael Biehl. Adaptive learning for complex-valued data. In Verleysen [125].

[32] M. Bdoiu, S. Har-Peled, and P. Indyk. Approximate clustering via core-sets. pages 250–257, 2002.

[33] Huanhuan Chen, Peter Tino, and Xin Yao. Probabilistic classification vector machines. *IEEE Transactions on Neural Networks*, 20(6):901–914, 2009.

[34] Weng Cho Chew and Lijun Jiang. Overview of large-scale computing: The past, the present, and the future. *Proceedings of the IEEE*, 101(2):227–241, 2013.

[35] R. Chitta, R. Jin, T.C. Havens, and A.K. Jain. Approximate kernel k-means: Solution to large scale kernel clustering. pages 895–903, 2011.

[36] F. S. Collins, M. Morgan, and A. Patrinos. The human genome project: Lessons from large-scale biology. *SCIENCE*, 300(5617):286–290, April 2003.

[37] B. Conan-Guez, F. Rossi, and A. El Golli. Fast algorithm and implementation of dissimilarity self-organizing maps. *Neural Networks*, 19(6-7):855–863, 2006.

[38] A. Cuzzocrea and M.M. Gaber. Data science and distributed intelligence: Recent developments and future insights. *Studies in Computational Intelligence*, 446:139–147, 2013.

[39] G. Da San Martino and A. Sperduti. Mining structured data. *IEEE Computational Intelligence Magazine*, 5(1):42–49, 2010.

[40] Hal Daumé, Jeff M. Phillips, Avishek Saha, and Suresh Venkatasubramanian. Protocols for learning classifiers on distributed data. *Journal of Machine Learning Research - Proceedings Track*, 22:282–290, 2012.

[41] W.J. Egan and S.L. Morgan. Outlier detection in multivariate analytical chemical data. *Analytical Chemistry*, 70(11):2372–2379, 1998.

[42] Terence A. Etchells and Paulo J. G. Lisboa. Orthogonal search-based rule extraction (osre) for trained neural networks: a practical and efficient approach. *IEEE Transactions on Neural Networks*, 17(2):374–384, 2006.

[43] Danyel Fisher, Rob DeLine, Mary Czerwinski, and Steven Drucker. Interactions with big data analytics. *interactions*, 19(3):50–59, May 2012.

[44] Shereen Fouad and Peter Tino. Adaptive metric learning vector quantization for ordinal classification. *Neural Computation*, 24(11):2825–2851, 2012.

[45] Neil Fraser. Neural network follies, march 2013.

[46] Brendan J. Frey and Delbert Dueck. Mixture modeling by affinity propagation. In *NIPS*, 2005.

[47] S. Gao, I.W. Tsang, and L.T. Chia. Sparse representation with kernels. *IEEE Transactions on Image Processing*, 22(2):423–434, 2013.

[48] N. Gianniotis and P. Tio. Visualization of tree-structured data through generative topographic mapping. *IEEE Transactions on Neural Networks*, 19(8):1468–1493, 2008.

[49] Inmar E. Givoni, Clement Chung, and Brendan J. Frey. Hierarchical affinity propagation. In Fabio Gagliardi Cozman and Avi Pfeffer, editors, *UAI*, pages 238–246. AUAI Press, 2011.

[50] Mohamed Farouk Abdel Hady, Friedhelm Schwenker, and Günther Palm. Semi-supervised learning for tree-structured ensembles of rbf networks with co-training. *Neural Networks*, 23(4):497–509, 2010.

[51] B. Hammer, A. Gisbrecht, and A. Schulz. How to visualize large data sets? *Advances in Intelligent Systems and Computing*, 198 AISC:1–12, 2013.

[52] B. Hammer and A. Hasenfuss. Topographic mapping of large dissimilarity data sets. *Neural Computation*, 22(9):2229–2284, 2010.

[53] B. Hammer, A. Micheli, A. Sperduti, and M. Strickert. A general framework for unsupervised processing of structured data. *Neurocomputing*, 57(1-4):3–35, 2004.

[54] B. Hammer, C. Saunders, and A. Sperduti. Special issue on neural networks and kernel methods for structured domains. *Neural Networks*, 18(8):1015–1018, 2005.

[55] B. Hammer, M. Strickert, and T. Villmann. On the generalization ability of grlvq networks. *Neural Processing Letters*, 21(2):109–120, 2005.

[56] Barbara Hammer, Andreas Rechtien, Marc Strickert, and Thomas Villmann. Rule extraction from self-organizing networks. In José R. Dorronsoro, editor, *ICANN*, volume 2415 of *Lecture Notes in Computer Science*, pages 877–883. Springer, 2002.

[57] Barbara Hammer and Thomas Villmann. Generalized relevance learning vector quantization. *Neural Networks*, 15(8-9):1059–1068, 2002.

[58] Trevor Hastie, Robert Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction: with 200 full-color illustrations*. New York: Springer-Verlag, 2001.

[59] T.C. Havens, J.C. Bezdek, C. Leckie, L.O. Hall, and M. Palaniswami. Fuzzy c-means algorithms for very large data. *Fuzzy Systems, IEEE Transactions on*, 20(6):1130 –1146, dec. 2012.

[60] Simon Haykin. *Neural Networks and Learning Machines (3rd Edition)*. Prentice Hall, 3 edition, November 2008.

[61] Haibo He, editor. *The 2012 International Joint Conference on Neural Networks (IJCNN), Brisbane, Australia, June 10-15, 2012*. IEEE, 2012.

[62] Tom Heskes. Self-organizing maps, vector quantization, and mixture modeling. *IEEE Transactions on Neural Networks*, 12(6):1299–1305, 2001.

[63] Jens Hocke, Kai Labusch, Erhardt Barth, and Thomas Martinetz. Sparse coding and selected applications. *KI*, 26(4):349–355, 2012.

[64] Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[65] Markus B. Huber, Kerstin Bunte, Mahesh B. Nagarajan, Michael Biehl, Lawrence A. Ray, and Axel Wismüller. Texture feature ranking with relevance learning to classify interstitial lung disease patterns. *Artificial Intelligence in Medicine*, 56(2):91–97, 2012.

[66] A. K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31:651–666, 2010.

[67] I. T. Jolliffe. *Principal Component Analysis*. Springer, second edition, October 2002.

[68] Kaggle. The Kaggle community database for data analysis contests, march 2013.

[69] M. Kästner, B. Hammer, M. Biehl, and T. Villmann. Functional relevance learning in generalized learning vector quantization. *Neurocomputing*, 90:85–95, 2012.

[70] M. Kastner, D. Nebel, M. Riedel, M. Biehl, and T. Villmann. Differentiable kernels in generalized matrix learning vector quantization. volume 1, pages 132–137, 2012.

[71] Marika Kästner and Thomas Villmann. Fuzzy supervised self-organizing map for semi-supervised vector quantization. In Leszek Rutkowski, Marcin Korytkowski, Rafal Scherer, Ryszard Tadeusiewicz, Lotfi A. Zadeh, and Jacek M. Zurada, editors, *ICAISC (1)*, volume 7267 of *Lecture Notes in Computer Science*, pages 256–265. Springer, 2012.

[72] Daniel A. Keim, Leishi Zhang, Milos Krstajic, and Svenja Simon. Solving problems with visual analytics: Challenges and applications. *JMPT*, 3(1):1–11, 2012.

[73] Sascha Klement and Thomas Martinetz. On the problem of finding the least number of features by l1-norm minimisation. In Timo Honkela, Wlodzislaw Duch, Mark A. Girolami, and Samuel Kaski, editors, *ICANN (1)*, volume 6791 of *Lecture Notes in Computer Science*, pages 315–322. Springer, 2011.

[74] T. Kohonen, M. R. Schroeder, and T. S. Huang, editors. *Self-Organizing Maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 3rd edition, 2001.

[75] Tim Kraska. Finding the needle in the big data systems haystack. *IEEE Internet Computing*, 17(1):84–86, 2013.

[76] Kai Labusch, Erhardt Barth, and Thomas Martinetz. Simple method for high-performance digit recognition based on sparse coding. *IEEE Transactions on Neural Networks*, 19(11):1985–1989, 2008.

[77] Kai Labusch, Erhardt Barth, and Thomas Martinetz. Robust and fast learning of sparse codes with stochastic gradient descent. *J. Sel. Topics Signal Processing*, 5(5):1048–1060, 2011.

[78] J.A. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. Information Science and Statistics. Springer, 2007.

[79] Jimmy Lin and Alek Kolcz. Large-scale machine learning at twitter. In K. Selçuk Candan, Yi Chen, Richard T. Snodgrass, Luis Gravano, and Ariel Fuxman, editors, *SIGMOD Conference*, pages 793–804. ACM, 2012.

[80] Paulo J. G. Lisboa, Alfredo Vellido, Roberto Tagliaferri, Francesco Napolitano, Michele Ceccarelli, José David Martín-Guerrero, and Elia Biganzoli. Data mining in cancer research [application notes]. *IEEE Comp. Int. Mag.*, 5(1):14–18, 2010.

[81] Ezequiel Lpez-Rubio. Probabilistic self-organizing maps for qualitative data. *Neural Networks*, 23(10):1208 – 1225, 2010.

[82] S. Madden. From databases to big data. *Internet Computing, IEEE*, 16(3):4 –6, may-june 2012.

[83] Thomas Martinetz and Klaus Schulten. Topology representing networks. *Neural Networks*, 7(3):507–522, 1994.

[84] E. Mwebaze, P. Schneider, F.-M. Schleif, J.R. Aduwo, J.A. Quinn, S. Haase, T. Villmann, and M. Biehl. Divergence based classification in learning vector quantization. *NeuroComputing*, 74:1429–1435, 2010.

[85] Sandra Ortega-Martorell, Paulo J. G. Lisboa, Alfredo Vellido, Margarida Julià-Sapé, and Carles Arús. Non-negative matrix factorisation methods for the spectral decomposition of mrs data from human brain tumours. *BMC Bioinformatics*, 13:38, 2012.

[86] Edgar Osuna, Robert Freund, and Federico Girosi. Improved training algorithm for support vector machines. pages 276–285, 1997.

[87] Clemens Otte. Safe and interpretable machine learning: A methodological review. In Christian Moewes and Andreas Nürnberger, editors, *Computational Intelligence in Intelligent Data Analysis*, volume 445 of *Studies in Computational Intelligence*, pages 111–122. Springer Berlin Heidelberg, 2013.

[88] John C. Platt. Fast training of support vector machines using sequential minimal optimization. pages 185–208. MIT Press, Cambridge, MA, USA, 1999.

[89] A. K. Qin and P. N. Suganthan. A novel kernel prototype-based learning algorithm. In *Proc. of ICPR'04*, pages 2621–624, 2004.

[90] Carl Edward Rasmussen. Gaussian processes for machine learning. MIT Press, 2006.

[91] A. Rebai, A. Joly, and N. Boujemaa. Blasso for object categorization and retrieval: Towards interpretable visual models. *Pattern Recognition*, 45(6):2377–2389, 2012.

[92] R. Riesen and H. Bunke. Graph classification based on vector space embedding. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(6):1053–1081, 2009.

[93] Héctor Ruiz, Ian H. Jarman, José D. Martín, Sandra Ortega-Martorell, Alfredo Vellido, Enrique Romero, and Paulo J. G. Lisboa. Towards interpretable classifiers with blind signal separation. In He [61], pages 1–7.

[94] T.H. Sarma, P. Viswanath, and B.E. Reddy. Speeding-up the kernel k-means clustering method: A prototype based hybrid approach. *Pattern Recognition Letters*, 34(5):564–573, 2013.

[95] Atsushi Sato and Keiji Yamada. Generalized learning vector quantization. In David S. Touretzky, Michael Mozer, and Michael E. Hasselmo, editors, *NIPS*, pages 423–429. MIT Press, 1995.

[96] Rafa Scherer. *Multiple Fuzzy Classification Systems*. Springer Publishing Company, Incorporated, 2012.

[97] F.-M. Schleif, B. Hammer, and Th. Villmann. Supervised neural gas for functional data and its application to the analysis of clinical proteom spectra. In *In Proceedings of the 9th International Work-Conference on Artificial Neural Networks (IWANN) 2007*, pages 1036–1044, Berlin, Heidelberg, Germany, 2007. Springer.

[98] F.-M. Schleif, T. Villmann, B. Hammer, and P. Schneider. Efficient kernelized prototype-based classification. *Journal of Neural Systems*, 21(6):443–457, 2011.

[99] Frank-Michael Schleif, Barbara Hammer, and Xibin Zhu. Semi-supervised vector quantization for proximity data. In *ESANN*, 2013.

[100] P. Schneider, M. Biehl, and B. Hammer. Distance learning in discriminative vector quantization. *Neural computation*, 21(10):2942–2969, 2009.

[101] P. Schneider, K. Bunte, H. Stiekema, B. Hammer, T. Villmann, and M. Biehl. Regularization in matrix relevance learning. *IEEE Transactions on Neural Networks*, 21(5):831–840, 2010.

[102] P. Schneider, T. Geweniger, F.-M. Schleif, M. Biehl, and T. Villmann. Multivariate class labeling in robust soft lvq. In *Proceedings of ESANN 2011*, pages 17–22, 2011.

[103] P. Schneider, F.-M. Schleif, T. Villmann, and M. Biehl. Generalized matrix learning vector quantizer for the analysis of spectral data. In Michel Verleysen, editor, *In Proceedings of the 16th European Symposium on Artificial Neural Networks (ESANN) 2008*, pages 451–456, Evere, Belgium, 2008. d-side publications.

[104] Petra Schneider, Michael Biehl, and Barbara Hammer. Distance learning in discriminative vector quantization. *Neural Computation*, 21(10):2942–2969, 2009.

[105] Bernhard Schölkopf, Alexander J. Smola, and Klaus R. Müller. Kernel principal component analysis. *Advances in kernel methods: support vector learning*, pages 327–352, 1999.

[106] Sambu Seo and Klaus Obermayer. Soft learning vector quantization. *Neural Computation*, 15(7):1589–1604, 2003.

[107] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9:371–421, 2008.

[108] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis and Discovery*. Cambridge University Press, 2004.

[109] Stephan Simmuteit, Frank-Michael Schleif, Thomas Villmann, and Thomas Elssner. Tanimoto metric in tree-som for improved representation of mass spectrometry data with an underlying taxonomic structure. In M. Arif Wani, Mehmed M. Kantardzic, Vasile Palade, Lukasz A. Kurgan, and Yuan Qi, editors, *ICMLA*, pages 563–567. IEEE Computer Society, 2009.

[110] N. Simon and R. Tibshirani. Standardization and the group lasso penalty. *Statistica Sinica*, 22(3):983–1001, 2012.

[111] P. Sun and X. Yao. Sparse approximation through boosting for learning large scale kernel machines. *IEEE Transactions on Neural Networks*, 21(6):883–894, 2010.

[112] David M. J. Tax and Robert P. W. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004.

[113] David M. J. Tax, Piotr Juszczak, Elzbieta Pekalska, and Robert P. W. Duin. Outlier detection using ball descriptions with adjustable metric. In Dit-Yan Yeung, James T. Kwok, Ana L. N. Fred, Fabio Roli, and Dick de Ridder, editors, *SSPR/SPR*, volume 4109 of *Lecture Notes in Computer Science*, pages 587–595. Springer, 2006.

[114] Markus Thom and Günther Palm. Sparse activity and sparse connectivity in supervised learning. *Journal of Machine Learning Research*, 14:1091–1143, 2013.

[115] T.S. Tian and G.M. James. Interpretable dimension reduction for classifying functional data. *Computational Statistics and Data Analysis*, 57(1):282–296, 2013.

[116] R. Tibshirani. Regression shrinkage and selection via the lasso: A retrospective. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 73(3):273–282, 2011.

[117] Peter Tiňo and Ian T. Nabney. Hierarchical gtm: Constructing localized nonlinear projection manifolds in a principled way. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):639–656, 2002.

[118] M.E. Tipping. The relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

[119] Ivor W. Tsang, James T. Kwok, and Pak-Ming Cheung. Core vector machines: Fast svm training on very large data sets. *Journal of Machine Learning Research*, 6:363–392, 2005.

[120] Ivor W. Tsang, James T. Kwok, and Kimo T. Lai. Core vector regression for very large regression problems. In Luc De Raedt and Stefan Wrobel, editors, *ICML*, volume 119 of *ACM International Conference Proceeding Series*, pages 912–919. ACM, 2005.

[121] I.W.-H. Tsang, A. Kocsor, and J.T.-Y. Kwok. Large-scale maximum margin discriminant analysis using core vector machines. *IEEE Transactions on Neural Networks*, 19(4):610–624, 2008.

[122] V.N. Vapnik. *The nature of statistical learning theory*. Statistics for engineering and information science. Springer, 2000.

[123] Alfredo Vellido, José D. Martín, Fabrice Rossi, and Paulo J. G. Lisboa. Seeing is believing: The importance of visualization in real-world machine learning applications. In *ESANN*, 2011.

[124] Alfredo Vellido, Jos D. Martn-Guerrero, and Paulo Lisboa. Making machine learning models interpretable. In Verleysen [125].

[125] Michel Verleysen, editor. *ESANN 2012, 20th European Symposium on Artificial Neural Networks, Bruges, Belgium, April 25-27, 2012, Proceedings*, 2012.

[126] T. Villmann and F.-M. Schleif. Functional vector quantization by neural maps. In *Proceedings of Whispers 2009*, page CD, 2009.

[127] Thomas Villmann, Ralf Der, J. Michael Herrmann, and Thomas Martinetz. Topology preservation in self-organizing feature maps: exact definition and measurement. *IEEE Trans. Neural Netw. Learning Syst.*, 8(2):256–266, 1997.

[128] V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World*. Springer, New York, 2005.

[129] Xiaoxia Wang, Peter Tiňo, and Mark A. Fardal. Multiple manifolds learning framework based on hierarchical mixture density model. In Walter Daelemans, Bart Goethals, and Katharina Morik, editors, *ECML/PKDD (2)*, volume 5212 of *Lecture Notes in Computer Science*, pages 566–581. Springer, 2008.

[130] Tim Winkler, Jens Drieseberg, Kai Hormann, Alexander Hasenfuss, and Barbara Hammer. Thinning mesh animations. In Oliver Deussen, Daniel A. Keim, and Dietmar Saupe, editors, *VMV*, pages 149–158. Aka GmbH, 2008.

[131] Zhirong Yang, Jaakko Peltonen, and Samuel Kaski. Scalable optimization of neighbor embedding for visualization. In *Proceedings of ICML 2013*, to appear.

[132] H. Yin. Nonlinear dimensionality reduction and data visualization: A review. *International Journal of Automation and Computing*, 4(3):294–303, 2007.

[133] E. Zare Borzeshi, M. Piccardi, K. Riesen, and H. Bunke. Discriminative prototype selection methods for graph embedding. *Pattern Recognition*, 46(6):1648–1657, 2013.

[134] K. Zhang and J.T. Kwok. Clustered nystrm method for large scale manifold learning and dimension reduction. *IEEE Transactions on Neural Networks*, 21(10):1576–1587, 2010.

[135] Xiaojin Zhu, Andrew B. Goldberg, Ronald Brachman, and Thomas Dietterich. *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers, 2009.