Prototype-based classification of dissimilarity data

Barbara Hammer, Bassam Mokbel,
Frank-Michael Schleif, and Xibin Zhu

CITEC centre of excellence, Bielefeld University, 33615 Bielefeld - Germany
{bhammer|bmokbel|fschleif|xzhu}@techfak.uni-bielefeld.de

Abstract. Unlike many black-box algorithms in machine learning, prototype based models offer an intuitive interface to given data sets since prototypes can directly be inspected by experts in the field. Most techniques rely on Euclidean vectors such that their suitability for complex scenarios is limited. Recently, several unsupervised approaches have successfully been extended to general possibly non-Euclidean data characterized by pairwise dissimilarities. In this paper, we shortly review a general approach to extend unsupervised prototype-based techniques to dissimilarities, and we transfer this approach to supervised prototype-based classification for general dissimilarity data.

1 Introduction

While machine learning techniques have revolutionized the possibility to deal with large and complex electronic data sets and highly accurate classification and clustering models can be inferred automatically from given data, many machine learning techniques have the drawback that they largely behave as black boxes. In consequence, applicants often have to simply 'trust' the output of such methods. It is in general hardly possible to see why an automatic classification method has taken a particular decision, nor is it possible to change the behavior or functionality of a given model from the outside due to the black box character. Hence many machine learning techniques are not suited to inspect large data sets in a meaningful human-understandable way.

Prototype-based methods represent their decisions in terms of typical representatives contained in the input space. Prototypes can directly be inspected by humans in the field in the same way as data points: for example, physicians can inspect prototypical medical cases, prototypical images can directly be displayed on the computer screen, prototypical action sequences of robots can be performed in a robotic simulation, etc. Since the decision in prototype-based techniques usually depends on the similarity of a given input to the prototypes stored in the model, a direct inspection of the taken decision in terms of the responsible prototype becomes possible.

Many different algorithms have been proposed in the literature which derive prototype based models from given data. Unsupervised techniques include

popular clustering algorithms such as simple k-means or fuzzy-k-means clustering, topographic mapping such as neural gas or the self-organizing map, and statistical counterparts such as the generative topographic mapping [19, 15, 3]. Supervised techniques take into account a priorly given class labeling and they try to find decision boundaries which accurately describe priorly known class labels. Popular methods include in particular different variants of learning vector quantization, some of which are derived from explicit cost functions or statistical models [15, 25, 28]. Besides different mathematical derivations of the models, these learning algorithms have in common that they arrive at sparse representations of data in terms of prototypical vectors, they form decisions based on the similarity of data to these prototypes, and their training is often very intuitive based on Hebbian principles. In addition to their direct interpretability, prototype-based models provide excellent generalization ability due to their sparse representation of data, see e.g. the work [11, 26] for explicit large margin generalization bounds for prototype-based techniques. Prototypes offer a compression and efficient representation of the important aspects of given data which very naturally allows to wrap the basic algorithms into an incremental life-long learning paradigm, treating the prototypes as a compact representation of all already seen data. This aspect has been used in diverse scenarios which deal with incremental settings or very large data sets, see e.g. [6, 14, 1].

One of the most severe restrictions of prototype-based methods is their dependency on the Euclidean distance and their restriction to Euclidean vector spaces only. This makes them unsuitable for complex or heterogeneous data sets: input features often have different relevance; further, high dimensionality easily disrupts the Euclidean norm due to accumulated noise in the data. This problem can partially be avoided by incorporating appropriate metric learning into the algorithms such as proposed e.g. in [26] or by looking at kernel versions of the techniques, see e.g. [24]. However, data in complex dynamical systems are often inherently non-Euclidean, such that an explicit or implicit representation in terms of Euclidean vectors is not possible at all. Rather, data have a complex structural form and dedicated dissimilarity measures should be used. Popular examples include dynamic time warping for time series, alignment for symbolic strings, graph or tree kernels for complex structures, the compression distance to compare sequences based on an information theoretic ground, and similar. These settings do not allow a vectorial representation of data at all, rather, data are given implicitly in terms of pairwise dissimilarities or relations; we refer to a 'relational data representation' in the following when addressing data sets which are represented implicitly by means of pairwise dissimilarities d_{ij} of data, D denotes the corresponding matrix of dissimilarities.

Recently, popular prototype-based clustering algorithms have been extended to deal with relational data. Since no embedding vector space is given a priori in these settings, the adaptation of prototypes by means of vectorial operations is no longer possible. One simple way around this problem is to restrict prototype positions to data positions. For techniques derived from a cost function, an optimization in the restricted feasible set of data positions leads to concrete

learning algorithms such as, for example, median clustering or affinity propagation [5, 7]. One drawback of this procedure consists in the fact that prototypes are very restricted if parts of the data space are sampled only sparsely, such that optimization is often very complicated. Due to this reason, the obtained accuracy can be severely reduced as compared to representations by prototypes in a continuous vector space.

In contrast, relational clustering implicitly embeds dissimilarity data in pseudo-Euclidean space and, this way, considers an implicit continuous update of prototypes for relational data which is equivalent to the standard setting in the Euclidean case, see e.g. [10]. An embedding in pseudo-Euclidean space is possible for every data set which is characterized by a symmetric matrix of pairwise dissimilarities [23], such that this approach covers a large number of relevant situations. This way, a highly improved flexibility of the prototypes is achieved since they can be represented in a smooth way independent of the sampling frequency of the data space. This approach has been integrated into unsupervised topographic mapping provided by neural gas, the self-organizing map, and the generative topographic mapping [10, 12, 9]. In all cases, a very flexible prototype-based data inspection technique for complex data sets which are described by pairwise dissimilarities arises. Example applications include the mapping of symbolic music data, large text data sets, or complex biomedical data sets such as mass spectra [21, 13, 8].

So far, the models proposed in the literature widely deal with unsupervised batch algorithms only. Supervised prototype-based classification for relational data described by pairwise dissimilarities has not yet been considered. The task of supervised classification occurs in diverse complex applications such as the classification of mass spectra according to the biomedical decision problem, the classification of environmental time series according to related toxicity, or the classification of music according to underlying composers or epochs. Supervised prototype-based techniques for general dissimilarity data would offer one striking possibility to arrive at human understandable classifiers in such settings.

In this contribution, we shortly review relational clustering algorithms for dissimilarity data and we propose a way to extend these techniques to supervised settings, arriving in particular at a relational extension of the popular supervised prototype-based learning vector quantization (LVQ) [15]. We derive an explicit algorithm based on a formalization of LVQ via a cost function [25, 28], and we test the accuracy of the approach in comparison to unsupervised alternatives in several benchmark scenarios. Based on the very promising accuracy achieved in these examples, we propose different extensions of the techniques to improve the sparsity, efficiency, and suitability to deal with large data sets.

2 Prototype based clustering and classification

Assume data $\boldsymbol{x}^i \in \mathbb{R}^n, i = 1, \ldots, m$, are given. Prototypes are elements $\boldsymbol{w}^j \in \mathbb{R}^n, j = 1, \ldots, k$, of the same space. They decompose data into receptive fields

$$R(\boldsymbol{w}^j) = \{\boldsymbol{x}^i \ : \ \forall k \ d(\boldsymbol{x}^i, \boldsymbol{w}^j) \leq d(\boldsymbol{x}^i, \boldsymbol{w}^k)\}$$

based on the squared Euclidean distance

$$d(\boldsymbol{x}^i, \boldsymbol{w}^j) = \|\boldsymbol{x}^i - \boldsymbol{w}^j\|^2 .$$

The goal of prototype-based machine learning techniques is to find prototypes which represent a given data set as accurately as possible.

In the unsupervised setting, the accuracy is often measured in terms of the accumulated distances of prototypes and data points in their receptive fields. Learning techniques can be derived from cost functions related to this objective. We exemplarily consider neural gas (NG) which constitutes a high-level technology to infer a prototype-based topographic mapping [19, 20]. NG is based on the objective

$$E_{\mathrm{NG}} = \sum_{i,j} \exp(-\mathrm{rk}(\boldsymbol{x}^i, \boldsymbol{w}^j)/\sigma^2) \cdot d(\boldsymbol{x}^i, \boldsymbol{w}^j)$$

where $\mathrm{rk}(\boldsymbol{x}^i, \boldsymbol{w}^j)$ denotes the rank of prototype $\boldsymbol{w}^j$, i.e. the number of prototypes which are closer to $\boldsymbol{x}^i$ than $\boldsymbol{w}^j$ measured according to the distance d. The parameter σ determines the degree of neighborhood cooperation. Batch optimization as introduced in [5] iteratively optimizes assignments and prototypes by means of the updates

$$\begin{array}{l} \text{compute } k_{ij} := \mathrm{rk}(\boldsymbol{x}^i, \boldsymbol{w}^j) \text{ for all } i \text{ and } j \text{ based on } d(\boldsymbol{x}^i, \boldsymbol{w}^j) \\ \text{set } w^j := \sum_i \exp(-k_{ij}/\sigma^2) \cdot \boldsymbol{x}^i / \sum_i \exp(-k_{ij}/\sigma^2) \text{ for all } j \end{array}$$

Starting from a random initialization, NG robustly determines prototype locations which represent data accurately as measured by the distances. In addition, the ranking of prototypes allows to infer the inherent data topology: prototypes are neighbored if and only if they are closest for at least one given data point [20].

In supervised settings, data $\boldsymbol{x}^i$ are equipped with prior class labels $c(\boldsymbol{x}^i) \in \{1, \ldots, L\}$ in a finite set of priorly known classes. An unsupervised prototype-based clustering gives rise to a classification by means of posterior labeling: a prototype $\boldsymbol{w}^j$ is assigned the label $c(\boldsymbol{w}^j)$ which corresponds to the majority of the labels of data points observed in its receptive field $R(\boldsymbol{w}^j)$. While this often yields astonishingly accurate classifiers, unsupervised training algorithms do not take into account the priorly known classes such that decision boundaries are not optimal. Learning vector quantization (LVQ) tries to avoid this problem by taking the labeling into account while positioning the prototypes [15]. Here we restrict to a variant of LVQ as proposed in [25], generalized LVQ (GLVQ), which has the benefit of a mathematical derivation from a cost function which can be related to the generalization ability of LVQ classifiers [26].

We assume every prototype is equipped with a label $c(\boldsymbol{w}^j)$ prior to training. The cost function of GLVQ is given as

$$E_{GLVQ} = \sum_i \Phi \left(\frac{d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) - d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))}{d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) + d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))} \right)$$

where Φ is a differentiable monotonic function such as the hyperbolic tangent, and $\boldsymbol{w}^+(\boldsymbol{x}^i)$ refers to the prototype closest to $\boldsymbol{x}^i$ with the same label as $\boldsymbol{x}^i$,

$\boldsymbol{w}^-(\boldsymbol{x}^i)$ refers to the closest prototype with a different label. This way, for every data point, its contribution to the cost function is small iff 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.

A learning algorithm can be derived thereof by means of a stochastic gradient descent. After a random initialization of prototypes, data $\boldsymbol{x}^i$ are presented in random order. Adaptation of the closest correct and wrong prototype takes place by means of the update rules

$$\begin{aligned}\Delta\boldsymbol{w}^+(\boldsymbol{x}^i) &\sim -\Phi'(\mu(\boldsymbol{x}^i)) \cdot \mu^+(\boldsymbol{x}^i) \cdot \nabla_{\boldsymbol{w}^+(\boldsymbol{x}^i)} d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) \\ \Delta\boldsymbol{w}^-(\boldsymbol{x}^i) &\sim \Phi'(\mu(\boldsymbol{x}^i)) \cdot \mu^-(\boldsymbol{x}^i) \cdot \nabla_{\boldsymbol{w}^-(\boldsymbol{x}^i)} d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))\end{aligned}$$

where

$$\mu(\boldsymbol{x}^i) = \frac{d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) - d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))}{d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) + d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))},$$

$$\mu^+(\boldsymbol{x}^i) = \frac{2 \cdot d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))}{(d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) + d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))^2},$$

and

$$\mu^-(\boldsymbol{x}^i) = \frac{2 \cdot d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)}{(d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) + d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))^2}.$$

For the squared Euclidean norm, the derivative yields

$$\nabla_{\boldsymbol{w}^j} d(\boldsymbol{x}^i, \boldsymbol{w}^j) = -(\boldsymbol{x}^i - \boldsymbol{w}^j),$$

leading to Hebbian update rules of the prototypes which take into account the priorly known class information, i.e. they adapt the closest prototypes towards / away from a given data point depending on the correctness of the classification. GLVQ constitutes one particularly efficient method to adapt the prototypes according to a given labeled data sets, alternatives such as techniques based on heuristics or algorithms derived from statistical models are possible [28, 27].

3 Dissimilarity data

Prototype-based techniques as introduced above are restricted to Euclidean vector spaces such that their suitability to deal with complex non-Euclidean data sets is highly limited. Since data are becoming more and more complex in many application domains e.g. due to improved sensor technology or dedicated data formats, the need to extend intuitive prototype-based techniques towards more general data has attracted some attention recently.

In the following, we assume that data $\boldsymbol{x}^i$ are not given as vectors, rather pairwise dissimilarities $d_{i,j} = d(\boldsymbol{x}^i, \boldsymbol{x}^j)$ of data points numbered i and j are available. D refers to the corresponding dissimilarity matrix. Note that it is easily possible to transfer similarities to dissimilarities and vice versa, see [23].

We assume symmetry $d_{ij} = d_{ji}$ and we assume $d_{ii} = 0$. However, we do not require that d refers to a Euclidean data space, i.e. D does not need to be embeddable in Euclidean space, nor does it need to fulfill the conditions of a metric.

One very simple possibility to transfer prototype-based models to this general setting is offered by a restriction of prototype positions. If we restrict $\boldsymbol{w}^j \in \mathcal{X} = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^m\}$ for all j, dissimilarities $d(\boldsymbol{x}^i, \boldsymbol{w}^j)$ are well defined and the cost functions of NG and GLVQ can be evaluated. Because of the discrete nature of the space of possible solutions, training can take place by means of an exhaustive search in $\mathcal{X}$ to find good prototype locations, in principle. This principle has been proposed to extend SOM and NG [5, 16]. One drawback of this technique is given by the restriction of flexibility of the prototypes on the one hand, and the complexity due to the exhaustive search, which is quadratic.

As an alternative, NG has been extended to so-called relational NG in [10]. Data described by pairwise dissimilarities can always be embedded in pseudo-Euclidean space, provided D is symmetric and has zero diagonal [23]. In pseudo-Euclidean space a symmetric bilinear form exists which induces the pairwise dissimilarities of data. As demonstrated in [10], NG can be performed in the pseudo-Euclidean vector space using this bilinear form. Since the embedding is usually not given explicitly and computation of an explicit embedding takes cubic complexity, the prototypes are usually adapted only implicitly based on the following observations: assume prototypes are represented as linear combinations of data points

$$\boldsymbol{w}^j = \sum_i \alpha_{ji} \boldsymbol{x}^i \text{ with } \sum_i \alpha_{ji} = 1 \,.$$

Then dissimilarities can be computed implicitly by means of the formula

$$d(\boldsymbol{x}^i, \boldsymbol{w}^j) = \|\boldsymbol{x}^i - \boldsymbol{w}^j\|^2 = [D \cdot \alpha_j]_i - \frac{1}{2} \cdot \alpha_j^t D \alpha_j$$

where $\alpha_j = (\alpha_{j1}, \ldots, \alpha_{jn})$ refers to the vector of coefficients describing the prototype $\boldsymbol{w}^j$ implicitly.

This way, batch adaptation of NG in pseudo-Euclidean space can be performed implicitly by means of the iterative adaptation:

compute $k_{ij} := \mathrm{rk}(\boldsymbol{x}^i, \boldsymbol{w}^j)$ based on $d(\boldsymbol{x}^i, \boldsymbol{w}^j) = [D \cdot \alpha_j]_i - \frac{1}{2} \cdot \alpha_j^t D \alpha_j$
set $\alpha_{ji} := \exp(-k_{ij}/\sigma^2) / \sum_i \exp(-k_{ij}/\sigma^2)$ for all j and i

This way, prototypes are represented implicitly by means of their coefficient vectors, and adaptation refers to the know pairwise dissimilarities d_{ij} only. We refer to relational NG (RNG) in the following. Initialization takes place by setting the coefficients to random vectors which sum up to 1. Note that the assumption $\sum_i \alpha_{ji} = 1$ is automatically fulfilled for optima of NG. Even for general settings, this assumption is quite reasonable since we can expect that the prototypes lie in the vector space spanned by the data.

For GLVQ, a kernelized version has been proposed in [24]. However, this refers to a kernel matrix only, i.e. it requires Euclidean similarities instead of

general symmetric dissimilarities. In particular, it must be possible to embed data in a possibly high dimensional Euclidean feature space. Here we transfer the ideas or RNG to GLVQ, obtaining a valid algorithm for general symmetric dissimilarities.

We assume that prototypes are given by linear combinations of data in pseudo-Euclidean space as before. Then, we can use the equivalent characterization of distances in the GLVQ cost function leading to the costs of relational GLVQ (RGLVQ):

$$E_{\mathrm{RGLVQ}} = \sum_i \Phi\left(\frac{[D\alpha^+]_i - \frac{1}{2}\cdot(\alpha^+)^t D\alpha^+ - [D\alpha^-]_i + \frac{1}{2}\cdot(\alpha^-)^t D\alpha^-}{[D\alpha^+]_i - \frac{1}{2}\cdot(\alpha^+)^t D\alpha^+ + [D\alpha^-]_i - \frac{1}{2}\cdot(\alpha^-)^t D\alpha^-}\right),$$

where as before the closest correct and wrong prototype are referred to, indicated by the superscript + and −, respectively. A stochastic gradient descent leads to adaptation rules for the coefficients α^+ and α^-: component k of these vectors is adapted by the rules

$$\Delta\alpha_k^+ \sim -\Phi'(\mu(\boldsymbol{x}^i))\cdot\mu^+(\boldsymbol{x}^i)\cdot\frac{\partial\left([D\alpha^+]_i - \frac{1}{2}\cdot(\alpha^+)^t D\alpha^+\right)}{\partial\alpha_k^+}$$

$$\Delta\alpha_k^- \sim -\Phi'(\mu(\boldsymbol{x}^i))\cdot\mu^-(\boldsymbol{x}^i)\cdot\frac{\partial\left([D\alpha^-]_i - \frac{1}{2}\cdot(\alpha^-)^t D\alpha^-\right)}{\partial\alpha_k^-}$$

where $\mu(\boldsymbol{x}^i)$, $\mu^+(\boldsymbol{x}^i)$, and $\mu^-(\boldsymbol{x}^i)$ are as above. The partial derivative yields

$$\frac{\partial[D\alpha_j]_i - \frac{1}{2}\cdot\alpha_j^t D\alpha_j}{\partial\alpha_{jk}} = d_{ik} - \sum_l d_{lk}\alpha_{jl}$$

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.

The resulting classifier represents clusters in terms of prototypes for general dissimilarity data. Although these prototypes correspond to vector positions in pseudo-Euclidean space, they can usually not be inspected directly because the pseudo-Euclidean embedding is not computed directly. Therefore, we use an approximation of the prototypes after training, substituting a prototype by its K nearest data points as measured by the given dissimilarity. To achieve a fast computation of this approximation, we enforce $\alpha_{ij} \geq 0$ during the updates.

Note that generalization of the classification to new data can be done in the same way as for RNG: given a novel data point $\boldsymbol{x}$ characterized by its pairwise dissimilarities $D(\boldsymbol{x})$ to the data used for training, the dissimilarity to the prototypes is given by $d(\boldsymbol{x}, \boldsymbol{w}^j) = D(\boldsymbol{x})^t\cdot\alpha_j - \frac{1}{2}\cdot\alpha_j^t D\alpha_j$. For an approximation of

prototypes by exemplars, obviously, only the dissimilarities to these exemplars have to be computed, i.e. a very sparse classifier results.

4 Experiments

We evaluate the algorithm for three benchmark data sets where data are characterized by pairwise dissimilarities:

1. The Copenhagen chromosomes data set constitutes a benchmark from cytogenetics [17]. A set of 4,200 human chromosomes from 22 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 [22].
2. The vibrio data set consists of 1,100 samples of vibrio bacteria populations characterized by mass spectra. The spectra contain 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 [18]. As usual, mass spectra display strong functional characteristics due to the dependency of subsequent masses, such that problem adapted similarities such as described in [2, 18] are beneficial. In our case, similarities are calculated using a specific similarity measure as provided by the BioTyper software[18].
3. The sonatas data set contains complex symbolic data similar to [21]. It is comprised of pairwise dissimilarities between 1,068 sonatas from the classical period (by Beethoven, Mozart and Haydn) and the baroque era (by Scarlatti and Bach). The musical pieces were given in the MIDI file format, taken from the online MIDI collection *Kunst der Fuge*[1]. Their mutual dissimilarities were measured with the normalized compression distance (NCD), see [4], using a specific preprocessing, which provides meaningful invariances for music information retrieval, such as invariance to pitch translation (transposition) and time scaling. This method uses a graph-based representation of the musical pieces to construct reasonable strings as input for the NCD, see [21]. The musical pieces are classified according to their composer.

These three data sets constitute typical examples of non-Euclidean data which occur in complex systems, such as medical image analysis, mass spectrometry, and symbolic domains. In all cases, dedicated preprocessing steps and dissimilarity measures for structures are used. The dissimilarity measures are inherently non-Euclidean and cannot be embedded isometrically in a Euclidean vector space.

We report the results of RNG in comparison to RGLVQ for these data sets. The number of prototypes is picked according to the number of priorly known classes, i.e. $k = 63$ for the chromosomes data (the smallest classes are represented by only one prototype), $k = 49$ for the vibrio data set, and $k = 5$ for

[1] `http://www.kunstderfuge.com`

	Chromosomes	Vibrio	Sonatas
RNG	0.911(0.004)	0.898(0.005)	0.745(0.002)
RNG ($K = 7$)	0.915(0.004)	0.993(0.004)	0.762(0.006)
RNG ($K = 5$)	0.912(0.003)	0.987(0.004)	0.754(0.008)
RNG ($K = 3$)	0.907(0.003)	0.976(0.004)	0.738(0.006)
RNG ($K = 1$)	0.893(0.002)	0.922(0.006)	0.708(0.004)
RGLVQ	0.927(0.002)	1.000(0.000)	0.839(0.002)
RGLVQ ($K = 7$)	0.923(0.005)	1.000(0.000)	0.794(0.005)
RGLVQ ($K = 5$)	0.917(0.001)	1.000(0.000)	0.788(0.006)
RGLVQ ($K = 3$)	0.912(0.002)	0.999(0.000)	0.780(0.009)
RGLVQ ($K = 1$)	0.902(0.000)	0.999(0.001)	0.760(0.008)

Table 1. Results of supervised and unsupervised prototype based classification for dissimilarity data on three different data sets. Evaluation is done by the classification accuracy measured in a repeated cross-validation. The standard deviation is given in parenthesis.

the sonatas data set. The prototypes are initialized randomly, and training is done for 5 epochs (chromosomes) or 10 epochs (vibrio, sonatas), respectively, starting from a random initialization of prototypes. The results are evaluated by the classification accuracy on the test set obtained in a repeated stratified 10-fold cross-validation with two repeats (for chromosomes) or ten repeats (vibrio, sonatas), respectively. The results are reported in Tab. 1, choosing different values K to approximate the prototypes by their nearest K exemplars.

In all cases, a good classification accuracy can be obtained using prototype-based relational data processing. Interestingly, the results obtained from trained RNG and RGLVQ classifiers using a K-approximation of the prototypes do not lead to much decrease of the classification accuracy, i.e. it is possible to represent the classes in terms of a small number of data points only. This way, the resulting classifiers can be directly inspected by experts in the field, because every class can be represented by a small number of exemplary data points.

In all cases, the incorporation of label information into the classifier leads to an increased classification accuracy of the resulting model, since priorly available information about class boundaries can better be taken into account in this setting. Thus, RGLVQ constitutes a very promising method to infer a high quality prototype-based classifier for general dissimilarity data sets which offers the possibility to inspect the clustering by directly referring to the prototypes or their approximation in terms of exemplars, respectively.

5 Conclusions

In this contribution, we have proposed a high quality supervised classification technique for general dissimilarity data which represents the decisions in the form of prototypes. Due to this representation, unlike many alternative black-box techniques, it offers the possibility of a direct inspection of the classifier

by humans. Further, unlike alternatives which are based on kernels such as the kernel GLVQ [24] or the relevance vector machine [29], the technique does not require that data are embeddable into Euclidean space, rather, a general symmetric dissimilarity matrix can be dealt with. Due to these properties, this technique seems very suited for interactive data inspection and modeling tasks, since it allows to deal with general dissimilarities (thus also very complex data or structural elements, or user-adapted dissimilarity values), and it allows an inspection by the user (including a direct change of the behavior by including or changing prototypes).

We demonstrated the accuracy of the technique for three different non-Euclidean data sets. More experimental evaluation and, in particular, an application to very large data sets are the subject of ongoing research. In particular, applications to spatiotemporal data as they occur in dynamic systems or robotics seem very interesting, where techniques such as temporal and spatial alignment offer very popular and powerful dissimilarity measures. Note that the extension of supervised prototype based classification to general dissimilarities by means of relational extensions is not restricted to GLVQ. Rather, alternative formulations based on cost functions such as soft robust LVQ as introduced in [28] can be extended in a similar way.

One central problem of relational classification as introduced above has not yet been considered in this contribution: while we arrive at sparse solutions by using approximations of prototypes, the techniques inherently possess quadratic complexity because of their dependency on the full dissimilarity matrix. This can lead to memory problems to store the full matrix, besides a long training time for large data sets. In applications, probably the biggest bottleneck is given by the necessity to compute the full dissimilarity matrix. In [8], two different ways to approximate the computation by linear time techniques which refer to an only linear subpart of the full dissimilarity matrix have been proposed and compared in the unsupervised setting: the classical Nyström approximation [30] to approximate the dissimilarity matrix by a low rank matrix, on the one side, and patch processing to compress the data consecutively by means of a prototype-based representation, on the other side. The approximation of RGLVQ by similar linear time techniques is a matter of ongoing research.

Acknowledgement Financial support from the Cluster of Excellence 277 Cognitive Interaction Technology funded in the framework of the German Excellence Initiative and from the "German Science Foundation (DFG)" under grant number HA-2719/4-1 is gratefully acknowledged.

References

1. Nikolai Alex, Alexander Hasenfuss, Barbara Hammer: Patch clustering for massive data sets. Neurocomputing 72(7-9): 1455-1469 (2009)
2. 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.
3. C. Bishop, M. Svensen, and C. Williams. The generative topographic mapping. *Neural Computation* 10(1):215-234, 1998.
4. R.Cilibrasi and M.B.Vitanyi, Clustering by compression, *IEEE Transactions on Information Theory* 51(4):1523-1545, 2005.
5. M. Cottrell, B. Hammer, A. Hasenfuss, and T. Villmann, "Batch and median neural gas," *Neural Networks*, vol. 19, pp. 762–771, 2006.
6. A. Denecke, H. Wersing, J.J. Steil, and E. Koerner. Online Figure-Ground Segmentation with Adaptive Metrics in Generalized LVQ. Neurocomputing 72(7-9):1470-1482 (2009)
7. B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, pp. 972–976, 2007.
8. Andrej Gisbrecht, Barbara Hammer, Frank-Michael Schleif and Xibin Zhu, Accelerating dissimilarity clustering for biomedical data analysis. Proceedings of SSCI 2011
9. Andrej Gisbrecht, Bassam Mokbel, Barbara Hammer: Relational generative topographic mapping. Neurocomputing 74(9): 1359-1371 (2011)
10. B. Hammer and A. Hasenfuss. Topographic Mapping of Large Dissimilarity Data Sets. Neural Computation 22(9):2229-2284, 2010.
11. B. Hammer and T. Villmann. Generalized relevance learning vector quantization. *Neural Networks*, 15(8-9):1059–1068, 2002.
12. Alexander Hasenfuss, Barbara Hammer: Relational Topographic Maps. IDA 2007: 93-105
13. A. Hasenfuss, W. Boerger, and B. Hammer. Topographic processing of very large text datasets. In C.H. Daglie and et al., editors, *Smart Systems Engineering: Computational Intelligence in Architecting Systes (ANNIE 2008)*, pages 525–532. ASME Press, 2008.
14. T. Kietzmann, S. Lange and M. Riedmiller. Incremental GRLVQ: Learning Relevant Features for 3D Object Recognition. Neurocomputing, 71 (13-15):28682879, Elsevier, 2008
15. T. Kohonen, editor. *Self-Organizing Maps*. Springer-Verlag New York, Inc., 3rd edition, 2001.
16. How to make large self-organizing maps for nonvectorial data, Teuvo Kohonen, Panu Somervuo, Neural Networks , vol. 15, no. 8-9, pp. 945-952, 2002
17. C. Lundsteen, J-Phillip, and E. Granum, "Quantitative analysis of 6985 digitized trypsin g-banded human metaphase chromosomes," *Clinical Genetics*, vol. 18, pp. 355–370, 1980.
18. T. Maier, S. Klebel, U. Renner, and M. Kostrzewa, "Fast and reliable maldi-tof ms–based microorganism identification," *Nature Methods*, no. 3, 2006
19. Martinetz, T.M., Berkovich, S.G., Schulten, K.J.: 'Neural-gas' network for vector quantization and its application to time-series prediction. IEEE Trans. on Neural Networks 4(4), 558–569 (1993)
20. Thomas Martinetz and Klaus Schulten, Topology representing networks, Neural Networks, Volume 7 Issue 3, 1994
21. Bassam Mokbel, Alexander Hasenfuss, Barbara Hammer: Graph-Based Representation of Symbolic Musical Data. GbRPR 2009: 42-51
22. M. Neuhaus and H. Bunke, "Edit distance based kernel functions for structural pattern classification," *Pattern Recognition*, vol. 39, no. 10, pp. 1852–1863, 2006.
23. E. Pekalska and R.P.W. Duin The Dissimilarity Representation for Pattern Recognition. Foundations and Applications. World Scientific, Singapore, December 2005.

24. Qin, A.K., Suganthan, P.N.: A novel kernel prototype-based learning algorithm. In: Proc. of ICPR'04. pp. 621–624 (2004)
25. 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.
26. P. Schneider, M. Biehl, and B. Hammer, "Adaptive relevance matrices in learning vector quantization," *Neural Computation*, vol. 21, no. 12, pp. 3532–3561, 2009.
27. Petra Schneider, Michael Biehl, Barbara Hammer: Distance Learning in Discriminative Vector Quantization. Neural Computation 21(10): 2942-2969 (2009)
28. Sambu Seo and Klaus Obermayer. Soft learning vector quantization. *Neural Computation*, 15(7):1589–1604, 2003.
29. Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. Journal of Machine Learning Research 1, 211-244
30. C. Williams and M. Seeger, "Using the nyström method to speed up kernel machines," in *Advances in Neural Information Processing Systems 13*. MIT Press, 2001, pp. 682–688.