Prototype based Fuzzy Classification in Clinical Proteomics

F.-M. Schleif [a,*]

[a] *University of Leipzig, Department of Mathematics and Computer Science, Institute of Computer Science, Leipzig, Germany*

T. Villmann [b]

[b] *University of Leipzig, Department of Medicine, Clinic for Psychotherapy, Leipzig, Germany*

B. Hammer [c]

[c] *Clausthal University of Technology, Department of Computer Science, Clausthal, Germany*

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.

Key words: 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 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 unsuper-

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

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

vised 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 $\mathbf{W} = \{\mathbf{w_r}\}$ be the set of all codebook vectors and $c_{\mathbf{r}}$ be the class label of $\mathbf{w_r}$. Furthermore, let $\mathbf{W}_c = \{\mathbf{w_r} | c_{\mathbf{r}} = c\}$ 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

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

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:

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

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 $d(\mathbf{v}_k, \mathbf{w}_\mathbf{r})$ denoted as $d^\lambda(\mathbf{v}_k, \mathbf{w}_\mathbf{r})$, which now is used in the soft assignments (2.3). One popular example is the scaled Euclidean metric

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

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

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

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 $\mathbf{w}_\mathbf{r}$ 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

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

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

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

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

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

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:

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

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 $k_j(\mathbf{v},\mathbf{W})$ with the derivative of the Heaviside-function $\theta(x)$ is the delta distribution $\delta(x)$. In this way one gets

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

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

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

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

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

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

	SNPC			SNG		SVM	
	train	test	$\bar{y^2}$	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

(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 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

	SNPC-R			LSNPC-R			SRNG		
	train	test	$\bar{y^2}$	train	test	$\bar{y^2}$	train	test	$\bar{y^2}$
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

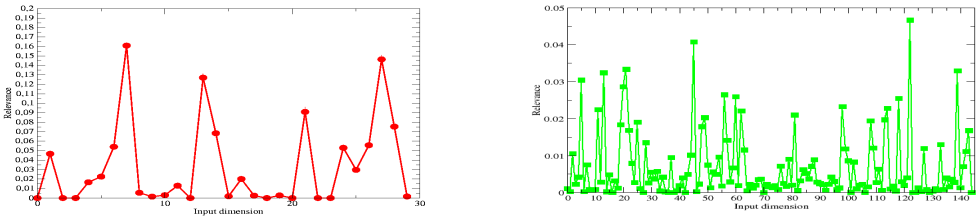

Fig. 1. Relevance profiles for the WBC (left) and LEUK (right) data set using SNPC-R

	FSNPC				FLNG			
	train	$\bar{y^2}$	test	$\bar{y^2}$	train	$\bar{y^2}$	test	$\bar{y^2}$
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

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 neighborhood cooperation.

Also for the fuzzy methods one can in general observe an improvement of the

	FSNPC-R				FLSNPC-R				FLNG-R			
	train	$\bar{y^2}$	test	$\bar{y^2}$	train	$\bar{y^2}$	test	$\bar{y^2}$	train	$\bar{y^2}$	test	$\bar{y^2}$
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. Mean square error for the labels ($\bar{y^2}$) are given for the training and test data.

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.

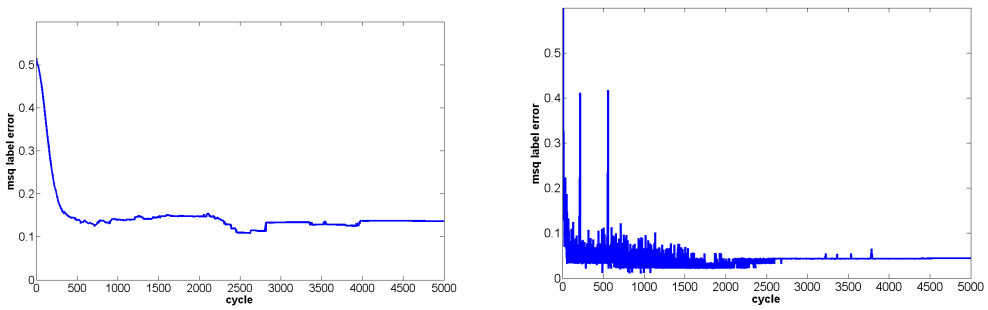

Fig. 2. Typical convergence curve for label error (LE) using FLNG-R (left) and FSNPC-R (right) for the WBC data. To get a more stable analysis the algorithms has been trained fix with 5000 cycles to obtain these LE curves using 6 prototypes.

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 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.