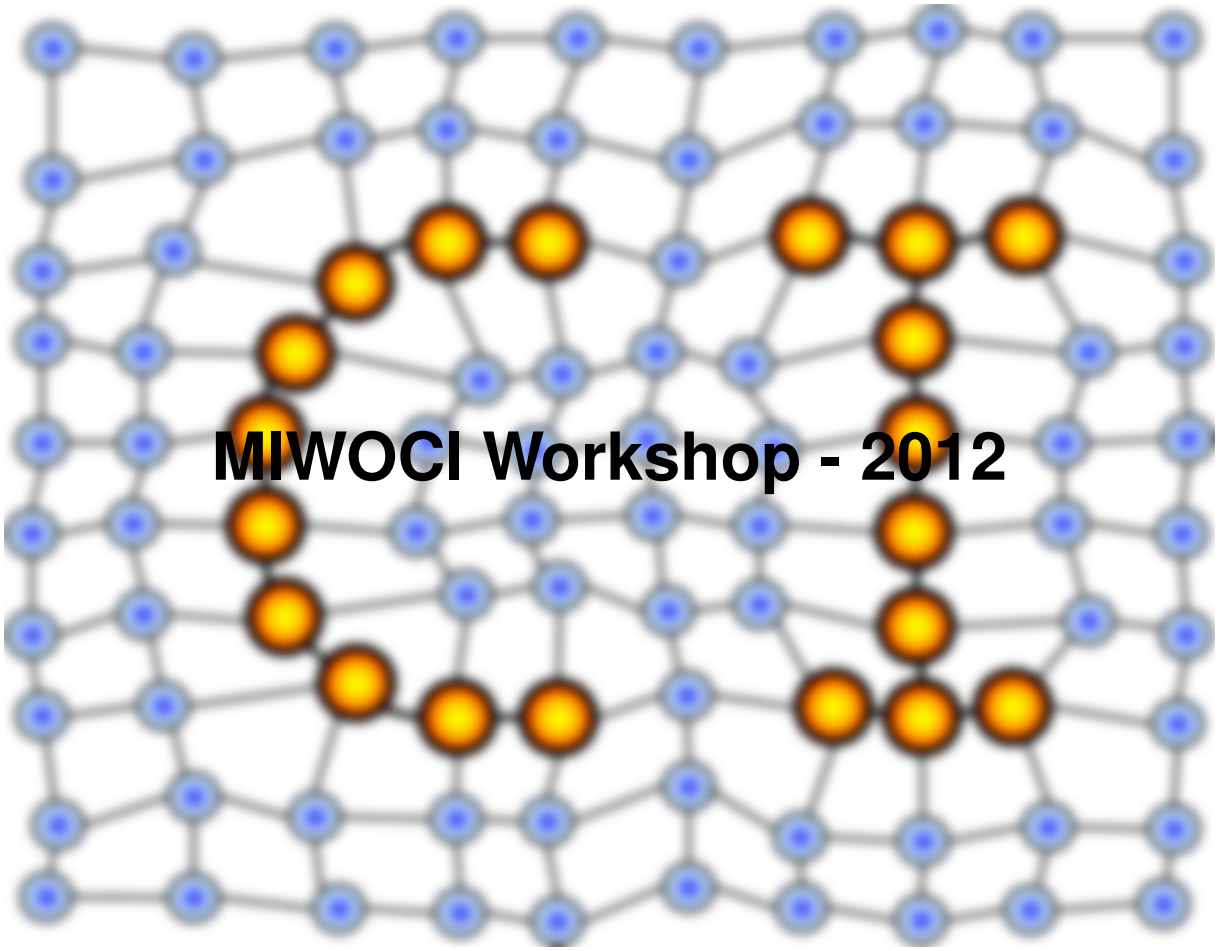
MACHINE LEARNING REPORTS



MIWOCI Workshop - 2012

Report 06/2012
Submitted: 18.10.2012
Published: 31.12.2012

Frank-Michael Schleif[1], Thomas Villmann[2] (Eds.)
(1) University of Bielefeld, Dept. of Technology CITEC - AG Computational Intelligence, Universitätsstrasse 21-23, 33615 Bielefeld
(2) University of Applied Sciences Mittweida, Technikumplatz 17, 09648 Mittweida, Germany

Figure 1: MiWoCi 2012

Contents

1 Fourth Mittweida Workshop on Computational Intelligence 4

2 Utilization of Correlation Measures in Vector Quantization for Analysis of Gene Expression Data 5

3 Border Sensitive Fuzzy Classification Learning in Fuzzy Vector Quantization 23

4 Class Border Sensitive Generalized Learning Vector Quantization 40

5 Accelerated Vector Quantization by Pulsing Neural Gas 57

Impressum

Publisher:	University of Applied Sciences Mittweida Technikumplatz 17, 09648 Mittweida, Germany
Editor:	Prof. Dr. Thomas Villmann Dr. Frank-Michael Schleif
Techical-Editor:	Dr. Frank-Michael Schleif
Contact:	fschleif@techfak.uni-bielefeld.de
URL:	http://techfak.uni-bielefeld.de/∼ fschleif/mlr/mlr.html
ISSN:	1865-3960

1 Fourth Mittweida Workshop on Computational Intelligence

From 02. Juli to 04 Juli 2012, 26 scientists from the University of Bielefeld, HTW Dresden, the Technical Univ. of Clausthal, Uni. of Groningen (NL), Univ. of Nijmegen (NL), Uni. of Paris 1 (F), the Fraunhofer Inst. for Factory Operation and Automation (IFF), the Fraunhofer Inst. for Applied Information Technology (FIT) and the Uni. of Applied Sciences Mittweida met in Mittweida, Germany, to continue the tradition of the Mittweida Workshops on Computational Intelligence - *MiWoCi'2012*. The aim was to present their current research, discuss scientific questions, and exchange their ideas. The seminar centered around topics in machine learning, signal processing and data analysis, covering fundamental theoretical aspects as well as recent applications, This volume contains a collection of extended abstracts.

Apart from the scientific merrits, this year's seminar came up with a few highlights which demonstrate the excellent possibilities offered by the surroundings of Mittweida. This year adventures were explored under intensive sunlight and very good weather conditions. The participants climbed to the high forests of Mittweida (Kletterwald) and enjoyed the exciting and fearing adventures provided on the top of the trees. Multiple jump offs from the *Wahnsinn* tour at a height of at least 20 meters were reported, but no participants were harmed. During a *wild water* journey (Paddeltour) the outstanding fitness of the researchers was demonstrated and some of them also demonstrated their braveness by swimming in the rapids followed by a nice barbecue.

Our particular thanks for a perfect local organization of the workshop go to Thomas Villmann as spiritus movens of the seminar and his PhD and Master students.

Bielefeld, December, 2012
Frank-M. Schleif

[1]E-mail: fschleif@techfak.uni-bielefeld.de
[2]University of Bielefeld, CITEC, Theoretical Computer Science, Leipzig, Germany

Utilization of Correlation Measures in Vector Quantization for Analysis of Gene Expression Data
– *A Review of Recent Developments* –

M. Kästner[1], M. Strickert[2], D. Labudde[3], M. Lange[1], S. Haase[1], and T. Villmann

[1]Computational Intelligence Group, University of Applied Sciences Mittweida, Technikumplatz 17, 09648 Mittweida, Germany

[2]Computational Intelligence Group, Philipps-University Marburg, Hans-Meerwein-Straße 6, 35032 Marburg , Germany

[3]Bioinformatics Group, University of Applied Sciences Mittweida, Technikumplatz 17, 09648 Mittweida, Germany

Gene expression data analysis is frequently performed using correlation measures whereas unsupervised and supervised vector quantization methods are usually designed for Euclidean distances. In this report we summarize recent approaches to apply correlation measures or divergences to those vector quantization algorithms for analysis of microarray gene expression data by Hebbian-like gradient descent learning.

*corresponding author, *email: thomas.villmann@hs-mittweida.de*

1 Introduction

Analysis and classification of gene expression data based on microarray data is still a challenging task in biology and medicine. It is one way of molecular biology and medicine to understand and to investigate biological processes, diseases and evolutions. As other diagnostic tools like mass spectrometry, microarrays deliver high-dimensional data to be analyzed. Prototype based methods have been established as powerful methods in high-dimensional data analysis in mass spectrometry with hundreds of spectral bands (data dimensions) [33, 34, 51]. In contrast to these data, microarray data consist up to thousands of gene expression levels whereas only a few data vectors are under consideration (tens or hundreds). Thus, the vector data space in gene expression analysis, one the one hand side, is more or less empty. This problem is also known as 'curse of dimensionality'. On the other hand, class differentiation in such high-dimensional data for only a few data points is trivial because a (linear) separation is practically always successful but, of course, not meaningful. Therefore, data preprocessing and gene selection is a crucial but challenging and complex task in gene expression data processing.

Usually, the data are preprocessed in advance. For example, the microarray data are investigated whether genes are expressed in similar way or not to reduce the dimensionality for subsequent data processing. For this purpose correlation analysis is a standard method frequently applied with subsequent selection based on correlation ranks [31]. Different strategies are applied: wrapper and filter techniques [16], classical discriminant analysis ranking [36] or visual analysis based on multi-dimensional scaling (MDS) [44]. Unsupervised approaches of dimension reduction like principal component analysis (PCA) [8, 13] or clustering techniques were also successfully applied to differentiate micro array gene expression data[3, 9]. Among them, neural network based PCA-methods and dimesion reduction techniques as well as feature selection methosd are very effective [45, 18, 24, 25].

One of the most successful approaches of unsupervised neural vector quantization and clustering is the self-organizing map [19]. Additionally to usual cluster and vector quantization schemes, SOMs offer great visualization abilities by inherently processing a non-linear data mapping onto a typically two-dimensional visualization grid which can be used for visual data inspection in case of topology preserving mappings [46]. However, the latter property is only obtained for certain conditions and has to be proven [47]. If only vector quantization accuracy is of interest without topology-preserving visualization the neural gas quantizer is better suited [22]. Supervised vector quantization for classification learning is mainly

influenced by the family of learning vector quantization algorithms (LVQ) [19]. Sato and Yamada extended this model such that an energy function reflecting the classification error is optimized by stochastic gradient learning (GLVQ) [32]. The GLVQ model can be further generalized including relevance or matrix learning learning for weighting the input dimensions or correlation between them, respectively [12, 35]. Yet, the resulting dissimilarity measure still remains a (weighted) Euclidean distance.

Most of the vector quantization algorithms have in common that the Euclidean distance is used for dissimilarity evaluation of the data and prototypes. For analysis of gene expression data, this can cause moderate problems: the Euclidean distance is sensitive to normalization like centralization and variance normalization, which would course problems when merging different data sets from different investigations [7, 2]. Therefore, other dissimilarity measures might be more qualified for gene expression analysis where frequently such problems occur. During the last years several alternatives were proposed. Among them, correlation based approaches seem to be very interesting for gene expression analysis [42, 39, 41, 40]. Other approaches use entropies and divergences [48, 52]. However, these approaches are fare away from application in standard vector quantization schemes for clustering. In this paper we consider the utilization of Pearson correlation for gene clustering by vector quantization as well as divergences.

In this paper we consider how correlation measures and divergences can be used in unsupervised and supervised vector quantization based on Hebbian-like gradient descent learning. In this way we provide a general framework for application of these neural vector quantization methods for the analysis of microarray gene expression data.

2 Unsupervised and Supervised Vector Quantization based on Cost Functions

Unsupervised neural vector quantizers have been established as powerful and robust methods for vector quantization frequently outperforming classic schemes like k-means. These approaches became standard tools for intuitive clustering of biological data [11, 51]. If only a few prototypes are used, vector quantizers can be seen as clustering algorithms where the prototypes are the cluster centers [21]. The basic principle of vector quantizers is to distribute prototype vectors in the data space as faithful as possible for adequate data representation. One of the most

prominent example is the self-organizing maps (SOMs) introduced by KOHONEN [19], which are also known for excellent visualization abilities for vectorial data. The strong performance and robustness may be dedicated to neighborhood learning paradigm based on Hebbian reinforcement learning [14], which is adopted from learning mechanisms in cortical areas of the brain. An alternative to SOM is the neural gas (NG, [22]) algorithm introduced by MARTINETZ, which combines the neighborhood learning idea for vector quantization with the theory of expanding (real) gases for the outstanding adaptive behavior but dropping the visualization ability.

Classification by neural vector quantization based on LVQ is the supervised counterpart of neural vector quantization by SOM and NG, heuristically motivated by KOHONEN [19]. A cost function based version keeping the basic ideas from LVQ is known as generalized LVQ (GLVQ) [32].

In the following we briefly review SOM, NG and GLVQ for being transferred to correlation and divergence measures lateron.

2.1 Basic Learning in SOM and NG

In the following we assume data $\mathbf{v} \in V \subseteq \mathbb{R}^n$ with data density $P(\mathbf{v})$ and a set of prototypes $W = \{\mathbf{w}_k\}_{k \in A} \subset \mathbb{R}^n$, where A is a finite index set. The reconstruction error is given in terms of the dissimilarity measure $d(\mathbf{v}, \mathbf{w}_k)$ between data and prototypes, which is assumed to be differentiable. Prototype adaptation in SOMs and NG can be realized as a stochastic gradient descent on a cost function E.[1] In that case, the gradient $\partial E / \partial \mathbf{w}_k$ contains the derivative $\partial d(\mathbf{v}, \mathbf{w}_k) / \partial \mathbf{w}_k$ originating from the chain rule of differentiation. In SOMs the index set A usually is a regular low-dimensional grid of rectangular or hexagonal shape. The indexes k are identified with the locations $\mathbf{r}_k$ in this grid, which are shortly denoted simply by $\mathbf{r}$. Then the lattice A is equipped with a distance d_A, which may be the shortest path counting each edge with weight one or the Euclidean distance after Euclidean embedding.

In particular, the cost function of the Heskes-variant of SOM is

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

[1] The SOM is equipped with a cost function only for the Heskes-variant [15].

with the so-called neighborhood function

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

and $d_A(\mathbf{r},\mathbf{r}') = \|\mathbf{r}-\mathbf{r}'\|_A$ is the distance in the SOM-lattice A according to its topological structure [15]. $K(\sigma)$ is a normalization constant depending on the neighborhood range σ. The symbol $\delta_{\mathbf{r}}^{s(\mathbf{v})}$ is the Kronecker and the wining neuron $s(\mathbf{v})$ is determined by

$$s(\mathbf{v}) = argmin_{\mathbf{r}\in A}\left(\sum_{\mathbf{r}'\in A} h_\sigma^{SOM}(\mathbf{r},\mathbf{r}')\cdot d(\mathbf{v},\mathbf{w}_{\mathbf{r}'})\right) \tag{2}$$

accordingt to [15]. Then the stochastic gradient prototype update for all prototypes is given as [15]:

$$\triangle\mathbf{w}_{\mathbf{r}} = -\varepsilon h_\sigma^{SOM}(\mathbf{r},s(\mathbf{v}))\frac{\partial d(\mathbf{v},\mathbf{w}_{\mathbf{r}})}{\partial\mathbf{w}_{\mathbf{r}}}. \tag{3}$$

depending on the derivatives of the used dissimilarity measure d in the data space.

In NG the dynamic neighborhood between prototypes for a given data vector $\mathbf{v}\in V$ is based on the winning rank of each prototype $\mathbf{w}_k$

$$rk_k(\mathbf{v}_i,W) = \sum_{l=1}^{N}\Theta\left(d(\mathbf{v}_i,\mathbf{w}_k)-d(\mathbf{v}_i,\mathbf{w}_l)\right) \tag{4}$$

where

$$\Theta(x) = \begin{cases} 0 & if \quad x\le 0 \\ 1 & else \end{cases} \tag{5}$$

is the Heaviside function [22]. The NG neighborhood function includes the ranks according to

$$h_\sigma^{NG}(k|\mathbf{v}) = c_\sigma^{NG}\cdot\exp\left(-\frac{(rk_k(\mathbf{v},W))^2}{2\sigma^2}\right) \tag{6}$$

with neighborhood range σ. Then the cost function is defined as

$$E_{\mathrm{NG}} = \sum_j\int P(\mathbf{v})\,h_\sigma^{NG}(rk_j(\mathbf{v},W))\,d(\mathbf{v},\mathbf{w}_j)\,d\mathbf{v} \tag{7}$$

with accompanying update

$$\triangle\mathbf{w}_j = -h_\sigma^{NG}(k_j(\mathbf{v},\mathbf{w}_j))\cdot\frac{\partial d(\mathbf{v},\mathbf{w}_j)}{\partial\mathbf{w}_j}. \tag{8}$$

It turns out that this rang-based vector quantization scheme is very robust and frequently delivers better results in terms of the mean squared error than other vector quantizers like SOM or k-means.

According to (3) and (8), both algorithms require differentiable dissimilarity measures $d(\mathbf{v}, \mathbf{w}_j)$. In case of the frequently applied squared Euclidean distance we immediately find

$$\frac{\partial d(\mathbf{v}, \mathbf{w}_j)}{\partial \mathbf{w}_j} = -2(\mathbf{v} - \mathbf{w}_j). \tag{9}$$

However, the concrete choice is left to the user under the assumption of minimum standards for general similarity measures like positive definiteness and reflexivity[26]. Hence, we have a great freedom do apply a task specific dissimilarity measure. Two appropriate choices for gene expression data are dressed in the next chapter.

2.2 Learning Vector Quantization by GLVQ

For supervised learning each training data vector $\mathbf{v} \in V \subset \mathbb{R}^n$ is equipped with a class label $x_\mathbf{v} \in \mathcal{C} = \{1, 2, 3, ..., C\}$. Now, the task is to distribute the set W of prototypes such that the classification error is minimized. For this purpose each prototype is also equipped with a class label $y_\mathbf{w}$ such that $\mathcal{C}$ is covered by all $y_\mathbf{w}$. After LVQ training a data point is assigned to the class of that prototype $\mathbf{w} \in W$ which has minimum distance.

A gradient based GLVQ scheme proposed by Sato and Yamada uses the following energy function:

$$E(W) = \frac{1}{2} \sum_{\mathbf{v} \in V} f(\mu_W(\mathbf{v})) \tag{10}$$

where the classifier function

$$\mu_W(\mathbf{v}) = \frac{d^+(\mathbf{v}) - d^-(\mathbf{v})}{d^+(\mathbf{v}) + d^-(\mathbf{v})} \tag{11}$$

approximates the non-differentiable classification error depending on W. The function $f : \mathbb{R} \to \mathbb{R}$ is monotonically increasing, usually chosen as sigmoid. Further, $d^+(\mathbf{v}) = d(\mathbf{v}, \mathbf{w}^+)$ denotes the distance between the data point $\mathbf{v}$ and the nearest prototype $\mathbf{w}^+$, which has the same label like $x_\mathbf{v} = y_{\mathbf{w}^+}$. In the following we abbreviate $d^+(\mathbf{v})$ simply by d^+. Analogously d^- is defined as the distance to the best prototype of all other classes.

The stochastic gradient learning for $E(W)$ is performed by

$$\frac{\partial_s E}{\partial \mathbf{w}^+} = \frac{\partial_s E}{\partial d^+(\mathbf{v})} \cdot \frac{\partial d^+(\mathbf{v})}{\partial \mathbf{w}^+}, \quad \frac{\partial_s E}{\partial \mathbf{w}^-} = \frac{\partial_s E}{\partial d^-(\mathbf{v})} \cdot \frac{\partial d^-(\mathbf{v})}{\partial \mathbf{w}^-} \tag{12}$$

with $\frac{\partial_s}{\partial}$ denotes the stochastic gradient and

$$\frac{\partial_s E}{\partial d^+(\mathbf{v})} = \frac{2d^-(\mathbf{v}) \cdot f'(\mu_W(\mathbf{v}))}{(d^+(\mathbf{v}) + d^-(\mathbf{v}))^2}, \quad \frac{\partial_s E}{\partial d^-(\mathbf{v})} = -\frac{2d^+(\mathbf{v}) \cdot f'(\mu_W(\mathbf{v}))}{(d^+(\mathbf{v}) + d^-(\mathbf{v}))^2}\,. \tag{13}$$

Obviously, in case of the (squared) Euclidean distance we have to calculate $\frac{\partial d^{\pm}(\mathbf{v})}{\partial \mathbf{w}^{\pm}}$ according to (9), which still refers to Hebbian-like learning. If more general dissimilarity measures $d(\mathbf{v}, \mathbf{w})$ are in use, the respective gradients have to be applied.

3 Appropriate Dissimilarity Measures for Gene Analysis in Microarrays Using Neural Vector Quantization Methods

As mentioned above, originally, the given models for supervised and unsupervised vector quantization were introduced using the Euclidean distance or generalizations thereof for dissimilarity calculations between data and prototype vectors. Recent developments investigate also other measures instead like divergences or kernel distances [23, 48, 49, 50]. Manual or semi-supervised gene selection in gene expression analysis is frequently based on the ranking of the correlations in microarrays [36]. Higher order correlations are taken into account if entropy based methods are applied in more sophisticated schemes like divergences [52]. In the following we review correlation measures and divergences for their use in vector quantization.

3.1 Pearson Correlation

Following the approach in [39], the *linear* Pearson correlation can be applied in gradient based vector quantization. Pearson correlation implicitly undertakes the data a centralization and therefore well suited for analysis of gene expression analysis [42, 43], where individually calibrated biomedical measuring devices are common [28, 42]. The Pearson correlation between a data vector $\mathbf{v} \in \mathbb{R}^n$ and a

prototype $\mathbf{w} \in \mathbb{R}^n$ is defined as

$$\varrho_P(\mathbf{v}, \mathbf{w}) = \frac{\sum_{k=1}^{n} (v_k - \mu_\mathbf{v}) \cdot (w_k - \mu_\mathbf{w})}{\sqrt{\sum_{k=1}^{n} (v_k - \mu_\mathbf{v})^2 \cdot \sum_{k=1}^{n} (w_k - \mu_\mathbf{w})^2}} \tag{14}$$

with $\mu_\mathbf{v}$ and $\mu_\mathbf{w}$ are the means of $\mathbf{v}$ and $\mathbf{w}$, respectively. Introducing the abbreviations $\mathcal{B} = \sum_{k=1}^{n} (v_k - \mu_\mathbf{v}) \cdot (w_k - \mu_\mathbf{w})$, $\mathcal{C} = \sum_{k=1}^{n} (v_k - \mu_\mathbf{v})^2$ and $\mathcal{D} = \sum_{k=1}^{n} (w_k - \mu_\mathbf{w})^2$ it can be rewritten as

$$\varrho_P(\mathbf{v}, \mathbf{w}) = \frac{\mathcal{B}}{\sqrt{\mathcal{C} \cdot \mathcal{D}}}. \tag{15}$$

The derivative $\frac{\partial \varrho_P(\mathbf{v},\mathbf{w})}{\partial \mathbf{w}}$ is obtained as

$$\frac{\partial \varrho_P(\mathbf{v}, \mathbf{w})}{\partial \mathbf{w}} = \varrho_P(\mathbf{v}, \mathbf{w}) \cdot \left(\frac{1}{\mathcal{B}} \mathbf{v} - \frac{1}{\mathcal{D}} \mathbf{w} \right) \tag{16}$$

paying attention to the fact that generally $\frac{\partial \varrho_P(\mathbf{v},\mathbf{w})}{\partial \mathbf{w}} \neq \frac{\partial \varrho_P(\mathbf{v},\mathbf{w})}{\partial \mathbf{v}}$ [40].

3.2 Soft Spearman Rank Correlation

Spearmans rank correlation is a *non-linear* correlation measure. However, due to its rank based computation scheme, it is not differentiable at hand. Now we will develop a soft version, which approximates the ranks using again the trick to describe the ranks in terms of sums of Heaviside functions (5) as in NG above: For that purpose we define an indicator matrix $\mathbf{R}(\mathbf{x})$ of a vector $\mathbf{x}$ as

$$\mathbf{R}(\mathbf{x}) = \begin{pmatrix} R(x_1, x_1) & \cdots & R(x_1, x_n) \\ \vdots & & \vdots \\ R(x_n, x_1) & \cdots & R(x_n, x_n) \end{pmatrix} \tag{17}$$

with

$$R(x_i, x_j) = \Theta(x_i - x_j). \tag{18}$$

The row vectors of the indicator matrix $\mathbf{R}(\mathbf{x})$ are denotes as $\mathbf{R}_i(\mathbf{x})$ determining the rank function

$$rnk(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{R}_i(\mathbf{x}). \tag{19}$$

Using this indicator matrix, the Spearman rank correlation between a data vector $\mathbf{v}$ and a prototype vector $\mathbf{w}$ can be expressed in terms of the Pearson correlation (14) by

$$\varrho_S(\mathbf{v}, \mathbf{w}) = \varrho_P(rnk(\mathbf{v}), rnk(\mathbf{w})) \tag{20}$$

using the rank vectors (19).

Unfortunately, the Spearman correlation $\varrho_S(\mathbf{v}, \mathbf{w})$ is not differentiable because of the indicator functions (18). A smoothed but differentiable version has to approximate the Heaviside function in (18). One possible parametrized solution applies the sigmoid Fermi function

$$f_\beta\left(\frac{x_i - x_j}{\sigma_\mathbf{x}}\right) = \frac{1}{1 + \exp\left(\frac{\beta(x_i - x_j)}{\sigma_\mathbf{x}}\right)} \tag{21}$$

such that the approximation for (18) becomes

$$R_\beta(x_i, x_j) = f_\beta\left(\frac{x_i - x_j}{\sigma_\mathbf{x}}\right) + \frac{1}{2} \tag{22}$$

with $\sigma_\mathbf{x}$ being the standard deviation of the vector $\mathbf{x}$ [1]. In the limit we have $\lim_{\beta\to\infty} R_\beta(x_i, x_j) = R(x_i, x_j)$.

In the following we use the estimators

$$\sigma_\mathbf{x} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_\mathbf{x})^2} \tag{23}$$

and $\mu_\mathbf{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$. The derivatives are obtained as

$$\frac{\partial R_\beta(x_i, x_j)}{\partial x_k} = f'_\beta\left(\frac{x_i - x_j}{\sigma_\mathbf{x}}\right) \cdot \frac{\beta}{\sigma_\mathbf{x}} \cdot \left[(\delta_{k,i} - \delta_{k,j}) - \frac{x_i - x_j}{\sigma_\mathbf{x}} \cdot \frac{\partial \sigma_\mathbf{x}}{\partial x_k}\right] \tag{24}$$

with

$$\frac{\partial \sigma_\mathbf{x}}{\partial x_k} = \frac{1}{\sigma_\mathbf{x}} \cdot \frac{1}{n} \cdot (x_k - \mu_\mathbf{x}) \tag{25}$$

as shown in the Appendix.

It turns out that this soft variant of Spearmans rank correlation can be related to other variants of soft and fuzzy rank correlations, the latter ones based on t-norms and t-conorms [1, 38]. Moreover, rank-based approaches frequently benefit from the robustness of this paradigm to achieve high performance.

3.3 Divergences

Higher order correlation can be taken into account using divergences or generalizations thereof. An overview over different types of divergences is given in [6, 4, 5]. Their use in gradient based vector quantization is extensively investigated in [48].

Clearly, divergences are closely related to entropy measures, which are considered to be useful also for gene ranking [52]. Therefore, we propose to consider also divergences in gene clustering based on microarray data to keep higher order correlations.

For that reason, we assume positive gene expression vectors $\mathbf{v}$ with expression levels $v_i \geq 0$. Equivalently the prototypes $\mathbf{w}$ are supposed to fulfill the condition $w_i > 0$. The generalized Kullback-Leibler-divergence (KLD) is defined as

$$D_{KL}(\mathbf{v}\|\mathbf{w}) = \sum_{i=1}^{n} v_i \ln\left(\frac{v_i}{w_i}\right) - (v_i - w_i) \tag{26}$$

with the derivatives

$$\frac{\partial D_{KL}(\mathbf{v}\|\mathbf{w})}{\partial w_i} = -\left(\frac{v_i}{w_i} - 1\right), \tag{27}$$

whereby we suppose the convention $0 \cdot \ln 0 = 0$ according to limit $\lim_{x\to 0} x \cdot \ln x = 0$ is valid.

An alternative to the KLD are *Rényi-divergences*

$$D_{\alpha}^{R}(\mathbf{v}\|\mathbf{w}) = \frac{1}{\alpha - 1} \log\left(\sum_{i=1}^{n} (v_i)^{\alpha} (w_i)^{1-\alpha} - \alpha \cdot v_i + (\alpha - 1) \cdot w_i + 1\right) \tag{28}$$

with $\alpha > 0$ [29],[30]. The Rényi-divergences converge to KLD for $\alpha \to 0$. Easy computation is achieved for $\alpha = 2$, which is well studied in information theoretic learning (ITL) by J. Principe [20, 27, 37].

A very robust divergence is the Cauchy-Schwarz-divergence (CSD) obtained from the more general γ-divergence

$$\frac{\sum_{k=1}^{n} (v_k - \mu_{\mathbf{v}}) \cdot (w_k - \mu_{\mathbf{w}})}{\sqrt{\sum_{k=1}^{n} (v_k - \mu_{\mathbf{v}})^2 \cdot \sum_{k=1}^{n} (w_k - \mu_{\mathbf{w}})^2}}$$

$$D_{\gamma}(\mathbf{v}\|\mathbf{w}) = \log\left[\frac{\left(\sum_{i=1}^{n} (v_i)^{\gamma+1}\right)^{\frac{1}{\gamma(\gamma+1)}} \cdot \left(\sum_{i=1}^{n} (w_i)^{\gamma+1}\right)^{\frac{1}{\gamma+1}}}{\left(\sum_{i=1}^{n} v_i (w_i)^{\gamma}\right)^{\frac{1}{\gamma}}}\right] \tag{29}$$

proposed by Fujisawa&Eguchi for the value $\gamma = 1$ [10, 17]. In that case, the γ-divergence becomes symmetric and the resulting CSD is a metric. Moreover, we remark that the relation

$$D_{\gamma=1}(\mathbf{v}\|\mathbf{w}) = -\log(\varrho_P(\mathbf{v}, \mathbf{w})) \tag{30}$$

with $\varrho_P(\mathbf{v}, \mathbf{w})$ being the Pearson correlation from (14) holds, if the data are centered. Yet, it turns out that frequently a value $\gamma \neq 1$ is optimal for a given vector quantization task [48].

4 Conclusion

Gene expression data analysis is frequently performed using correlation measures. In this report we summarized recent approaches in unsupervised and supervised neural vector quantization algorithms using several kinds of correlation measures including Pearson and Spearman correlations. We point out, how these dissimilarity measures, which have shown to be successful in microarray gene expression data analysis, can easily be plugged in into well-known algorithms like self-organizing maps and learning vector quantization.

Appendix

We consider the derivative $\frac{\partial \sigma_{\mathbf{v}}}{\partial v_k}$ of the standard deviation $\sigma_{\mathbf{v}} = \sqrt{\frac{1}{n} S}$ with $S = \sum_{i=1}^{n} (v_i - \mu_{\mathbf{v}})^2$ as equivalent for (23). Then the derivative becomes

$$\frac{\partial \sigma_{\mathbf{v}}}{\partial v_k} = \frac{1}{2\sigma_{\mathbf{v}} \cdot n} \cdot \frac{\partial S}{\partial v_k}. \tag{31}$$

We investigate $\frac{\partial S}{\partial v_k}$, which yields

$$\begin{aligned} \frac{\partial S}{\partial v_k} &= \sum_{i=1}^{n} \frac{\partial (v_i - \mu_{\mathbf{v}})^2}{\partial v_k} \\ &= 2 \sum_{i=1}^{n} (v_i - \mu_{\mathbf{v}}) \frac{\partial (v_i - \mu_{\mathbf{v}})}{\partial v_k} \\ &= 2 \sum_{i=1}^{n} (v_i - \mu_{\mathbf{v}}) \left[\delta_{i,k} - \frac{\partial \mu_{\mathbf{v}}}{\partial v_k} \right]. \end{aligned} \tag{32}$$

Now we have to take into account the estimate $\mu_{\mathbf{v}} = \frac{1}{n} \sum_{l=1}^{n} v_i$ which delivers

$$\frac{\partial \mu_{\mathbf{v}}}{\partial v_k} = \frac{1}{n} \sum_{l=1}^{n} \delta_{k,l} = \frac{1}{n}. \tag{33}$$

Putting the pieces together we get

$$\frac{\partial S}{\partial v_k} = 2 \cdot \left[(v_k - \mu_{\mathbf{v}}) - \frac{1}{n} \sum_{i=1}^{n} (v_i - \mu_{\mathbf{v}}) \right]. \tag{34}$$

Now we consider

$$\begin{aligned} \frac{1}{n}\sum_{i=1}^{n}(v_i - \mu_{\mathbf{v}}) &= \frac{1}{n}\sum_{i=1}^{n} v_i - \frac{1}{n}\sum_{i=1}^{n}\mu_{\mathbf{v}} \\ &= \mu_{\mathbf{v}} - \frac{1}{n}\cdot n \cdot \mu_{\mathbf{v}} \\ &= 0 \end{aligned}$$

and, hence,

$$\frac{\partial \sigma_{\mathbf{v}}}{\partial v_k} = \frac{1}{n \cdot \sigma_{\mathbf{v}}} \cdot (v_k - \mu_{\mathbf{v}}) \tag{35}$$

is finally obtained.

References

[1] U. Bodenhofer and F. Klawonn. Robust rank correlation coeffcients on the basis of fuzzy orderings: Initial steps. *Mathware & Soft Computing*, 15:5–20, 2008.

[2] S. Chelloug, S. Meshoul, and M. Batouche. Clustering microarray data within amorphous computing paradigm and growing neural gas algorithm. In M. Alia and R. Dapoigny, editors, *Advances in Applied Artificial Intelligence*, number 4031 in Lecture Notes in Computer Science (LNCS), pages 809–818. Springer, 2006.

[3] F. Chiaromonte and J. Martinelli. Dimension reduction strategies for analyzing global gene expression data with a response. *Mathematical Biosciences*, 176:123–144, 2002.

[4] A. Cichocki and S.-I. Amari. Families of alpha- beta- and gamma- divergences: Flexible and robust measures of similarities. *Entropy*, 12:1532–1568, 2010.

[5] A. Cichocki, S. Cruces, and S.-I. Amari. Generalized alpha-beta divergences and their application to robust nonnegative matrix factorization. *Entropy*, 13:134–170, 2011.

[6] A. Cichocki, R. Zdunek, A. Phan, and S.-I. Amari. *Nonnegative Matrix and Tensor Factorizations*. Wiley, Chichester, 2009.

[7] D. Covell, A. Wallqvist, A. Rabow, and N. Thanki. Molecular classification of cancer: unsupervised self-organizing map analysis of gene expression microarray data. *Molecular cancer therapeutics*, 2(36):317–332, 2003.

[8] J. F. P. da Costa, H. Alonso, and L. Roque. A weighted principal component analysis and its application to gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(1):246–252, 2011.

[9] J. Dai and L.Lieu. Dimension reduction for classification with gene expression microarray data. *Statistical Applications in Genetics and Molecular Biology*, 5(1):1–19, 2006.

[10] H. Fujisawa and S. Eguchi. Robust parameter estimation with a small bias against heavy contamination. *Journal of Multivariate Analysis*, 99:2053–2081, 2008.

[11] B. Hammer, A. Hasenfuß, F.-M. Schleif, T. Villmann, M. Strickert, and U. Seiffert. Intuitive clustering of biological data. In *Proceedings of the International Joint Conference on Artificial Neural Networks (IJCNN 2007)*, pages 1877–1882, 2007.

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

[13] X. Han. Nonnegative principal component analysis for cancer molecular pattern discovery. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(3):537–549, 2010.

[14] D. Hebb. *The Organization of Behavior. A Neuropsychological Theory*. John Wiley, New York, 1949.

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

[16] I. Inza, P. L. naga, R. Blanco, and A. Cerrolaza. Filter versus wrapper gene selection approaches in DNA microarray domains. *Artificial Intelligence in Medicine*, 31(2):91–103, 2004.

[17] R. Jenssen, J. Principe, D. Erdogmus, and T. Eltoft. The Cauchy-Schwarz divergence and Parzen windowing: Connections to graph theory and Mercer kernels. *Journal of the Franklin Institute*, 343(6):614–629, 2006.

[18] S. Kaski. SOM-based exploratory analysis of gene expression data. In N. Allinson, H. Yin, L. Allinson, and J. Slack, editors, *Advances in Self-Organising Maps*, pages 124–31. Springer, 2001.

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

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

[21] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28:84–95, 1980.

[22] 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.

[23] E. Mwebaze, P. Schneider, F.-M. Schleif, J. Aduwo, J. Quinn, S. Haase, T. Villmann, and M. Biehl. Divergence based classification in learning vector quantization. *Neurocomputing*, 74(9):1429–1435, 2011.

[24] E. Oja. New aspects on the subspace methods of pattern recognition. In *Electron. Electr. Eng. Res. Stud. Pattern Recognition and Image Processing Ser. 5*, pages 55–64. Letchworth, UK, 1984.

[25] E. Oja. Neural networks—advantages and applications. In C. Carlsson, T. Järvi, and T. Reponen, editors, *Proc. Conf. on Artificial Intelligence Res. in Finland*, number 12 in Conf. Proc. of Finnish Artificial Intelligence Society, pages 2–8, Helsinki, Finland, 1994. Finnish Artificial Intelligence Society.

[26] E. Pekalska and R. Duin. *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications*. World Scientific, 2006.

[27] J. Principe. *Information Theoretic Learning*. Springer, Heidelberg, 2010.

[28] G. Raghava and J. H. Han. Correlation and prediction of gene expression level from amino acid and dipeptide composition of its protein. *BMC Bioinformatics*, 6:59, 2005.

[29] A. Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, 1961.

[30] A. Rényi. *Probability Theory*. North-Holland Publishing Company, Amsterdam, 1970.

[31] Y. Saeys, I. Inza, and P. Larra-Naga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

[32] 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.

[33] F.-M. Schleif, T. Villmann, and B. Hammer. Prototype based fuzzy classification in clinical proteomics. *International Journal of Approximate Reasoning*, 47(1):4–16, 2008.

[34] F.-M. Schleif, T. Villmann, M. Kostrzewa, B. Hammer, and A. Gammerman. Cancer informatics by prototype networks in mass spectrometry. *Artificial Intelligence in Medicine*, 45(2-3):215–228, 2009.

[35] P. Schneider, B. Hammer, and M. Biehl. Distance learning in discriminative vector quantization. *Neural Computation*, 21:2942–2969, 2009.

[36] A. Sharma and K. Paliwal. Cancer classification by gradient LDA technique using microarray gene expression data. *Data & Knowledge Enginneering*, 66:338–347, 2008.

[37] A. Singh and J. Principe. Information theoretic learning with adaptive kernels. *Signal Processing*, 91(2):203–213, 2011.

[38] M. Strickert. Enhancing M|G|RLVQ by quasi step discriminatory functions using 2^{nd} order training. *Machine Learning Reports*, 5(MLR-06-2011):5–15, 2011. ISSN:1865-3960, http://www.techfak.uni-bielefeld.de/ ~ fschleif/mlr/mlr_06_2011.pdf.

[39] M. Strickert, F.-M. Schleif, U. Seiffert, and T. Villmann. Derivatives of pearson correlation for gradient-based analysis of biomedical data. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, (37):37–44, 2008.

[40] M. Strickert, F.-M. Schleif, T. Villmann, and U. Seiffert. Unleashing pearson correlation for faithful analysis of biomedical data. In M. Biehl, B. Hammer, M. Verleysen, and T. Villmann, editors, *Similarity-based Clustering*, volume 5400 of *LNAI*, pages 70–91. Springer, Berlin, 2009.

[41] 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.

[42] M. Strickert, N. Sreenivasulu, B. Usadel, and U. Seiffert. Correlation-maximizing surrogate gene space for visual mining of gene expression patterns in developing barley endosperm tissue. *BMC*, 8:165, 2007.

[43] M. Strickert, N. Sreenivasulu, T. Villmann, and B. Hammer. Robust centroid-based clustering using derivatives of Pearson correlation. In P. Encarnação and A. Veloso, editors, *Proceedings of the First International Conference on Biomedical Electronics and Devices, BIOSIGNALS 2008*, volume 2, pages 197–203, Funchal, Madeira, Portugal, 2008. INSTICC - Institute for Systems and Technologies of Information, Control and Communication.

[44] Y.-H. Taguchi and Y. Oono. Relational patterns of gene expression via non-metric multidimensional scaling analysis. *BMC Bioinformatics*, 21(6):730–740, 2005.

[45] K. Torkkola, R. M. Gardner, T. Kaysser-Kranich, and C. Ma. Self-organizing maps in mining gene expression data. In J. R. Gattiker, J. T. L. Wang, and P. P. Wang, editors, *Information Sciences*, volume 139, pages 79–96. Motorola Labs, MD ML28, 2001.

[46] J. Vesanto. Som-based data visualization methods. *Intelligent Data Analysis*, 3:111–26, 1999.

[47] T. Villmann, R. Der, M. Herrmann, and T. M. Martinetz. Topology preservation in self-organizing feature maps: exact definition and measurement. *IEEE Transactions on Neural Networks*, 8(2):256–266, 1997.

[48] T. Villmann and S. Haase. Divergence based vector quantization. *Neural Computation*, 23(5):1343–1392, 2011.

[49] T. Villmann and S. Haase. A note on gradient based learning in vector quantization using differentiable kernels for Hilbert and Banach spaces. *Machine Learning Reports*, 6(MLR-02-2012):1–29, 2012. ISSN:1865-3960, http://www.techfak.uni-bielefeld.de/~fschleif/mlr/mlr_02_2012.pdf.

[50] T. Villmann, S. Haase, and M. Kästner. Gradient based learning in vector quantization using differentiable kernels. In P. Estevez, J. Principe, and P. Zegers, editors, *Advances in Self-Organizing Maps: 9th International Workshop WSOM 2012 Santiage de Chile*, volume 198 of *Advances in Intelligent Systems and Computing*, pages 193–204, Berlin, 2012. Springer.

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

[52] S. Zhu, D. Wang, K. Yu, T. Li, and Y. Gong. Feature selection for gene expression using model-based entropy. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(1):25–36, 2010.

Border Sensitive Fuzzy Classification Learning in Fuzzy Vector Quantization

T. Villmann* , T. Geweniger, and M. Kästner

Computational Intelligence Group, University of Applied Sciences Mittweida,
Technikumplatz 17, 09648 Mittweida, Germany

Abstract

Unsupervised fuzzy vector quantizers like fuzzy-c-means are widely applied in the analysis of high-dimensional data and show a robust behavior. However, those models also are applied for classification tasks. In that case each class is independently learned by a separate model. In this paper we overcome this disadvantage. By incorporation of neighborhood relations the models interact while the neighborhood is dynamic. In consequence, the prototypes are not longer distributed in the inner areas of the class distribution but placed close to the borders. Thus, a border sensitive fuzzy classification scheme is obtained.

1 Introduction

The utilization of unsupervised prototype based vector quantization methods to solve classification tasks is a common strategy: For example, unsupervised vector quantization algorithms are adapted to serve also as classifiers [6, 11, 18, 44, 45]. Another strategy is to apply several different unsupervised vector quantizers particularly dedicated to be responsible for certain classes, which may interact [11, 42]. Obviously, these strategies can also be applied for fuzzy clustering and classification [36, 41].

*corresponding author, *email: thomas.villmann@hs-mittweida.de*

Special interest is frequently given to the knowledge of decision borders between classes. This problem is explicitly addressed in support vector machine learning, which determines so-called support vectors approximating and indicating the borders between the classes [34, 30]. Recently, the idea of emphasizing the class borders while training several instances of unsupervised (fuzzy-) c-means algorithms (FCMs) based on the Euclidean distance is provided [47]. Several instances interact with each other while learning the classification task. However, beside FCM there exist other variants of unsupervised fuzzy vector quantizers realizing fuzzy probabilistic or fuzzy possibilistic models, which could be used in this communication model. These algorithms have in common that they do not incorporate neighborhood cooperativeness between the prototypes, and, therefore, may get stuck in local optima. Fortunately, neighborhood cooperativeness can be incorporated in these models such that stucking in local optima can be avoided. Another aspect of the family of fuzzy vector quantizers is that these algorithms are frequently based on the Euclidean distance to judge the data dissimilarity. In contrast, support vector machines (SVMs,[30]) implicitly map the data into a high-dimensional (maybe infinite dimensional) function Hilbert space determining their dissimilarity just in this Hilbert space based on kernel distances instead in the original data space [35]. It turns out that this space offers a rich topological structure such that respective classifications become very efficient with high accuracy. Recently, it was pointed out that this framework can be transferred to the case that the original data are preserved but equipped with a new metric. This is equivalent to the functional Hilbert space used in SVMs when applying differentiable kernels [41]. Obviously, this idea could be transferred to fuzzy vector quantizers, as well.

In this paper we propose a combination of differentiable kernel vector quantization with the idea of neighborhood cooperativeness for fuzzy vector quantization. Here, we generalize the idea of class border sensitive prototype adaptation of combined unsupervised fuzzy vector quantizers based on nearest neighbors as presented in [47] to neighborhood oriented learning. As well as we emphasize neighborhood oriented learning for the prototypes in each of the class responsible unsupervised fuzzy vector quantizers. In result we end-up with a robust supervised fuzzy classification network combining several unsupervised neighborhood cooperativeness incorporating fuzzy vector quantizers, which do not act independently but sensitized to class borders again stressing the idea of neighborhood cooperativeness for class border detection to avoid local optima.

The outline of the paper is as follows: First, we briefly review different unsuper-

vised fuzzy vector quantizers and show how differentiable kernels can be applied in these approaches. Second, we recognize the idea of combining several interacting instances of such unsupervised fuzzy vector quantizers for classification while sensitizing them to be particularly responsible for the class borders according to the idea presented in [36, 47]. In the next step we incorporate the idea of neighborhood cooperativeness on both levels: within several unsupervised fuzzy vector quantizers as well as in the class border sensitive interaction. Finally, we discuss the model using kernel distances as dissimilarity measure.

2 Unsupervised Fuzzy-Probabilistic, Fuzzy-Possibilistic and Soft Vector Quantization

We start by reviewing the basic principles and algorithms for unsupervised fuzzy vector quantization. We assume a data set $V = \{\mathbf{v}_i\}_{i=1}^{N} \subseteq \mathbb{R}^n$ and a set $W = \{\mathbf{w}_k\}_{k=1}^{M} \subset \mathbb{R}^n$ of prototypes. Further, we suppose a distance measure $d_{i,k} = d(\mathbf{v}_i, \mathbf{w}_k)$, which implies that the data space is equipped with a semi-inner product. Frequently the distance is chosen as the Euclidean distance corresponding to the Euclidean inner product.

The most prominent fuzzy vector quantization algorithm is the Fuzzy-c-Means (FCM) [1, 8], which is the fuzzy generalization of the standard c-means algorithm [7, 24]. Many variants are proposed, e.g. for relational data [2], median clustering [10], possibilistic variants [20, 21, 26], or using several kinds of dissimilarities like divergences [14, 40] or kernels [13]. Integration of neighborhood cooperative nessaccording to the crisp neural gas vector quantizer (NG, [25]) is studied in [38] resulting fuzzy neural gas (FNG).

2.1 Basic Fuzzy Vector Quantizers

We consider the most general variant (known as PFCM)

$$E_{PFCM}(\mathbf{U}, V, W, \delta, \gamma) = \sum_{k=1}^{M} \sum_{i=1}^{N} \left(\gamma \cdot u_{i,k}^{m} + (1-\gamma) \cdot t_{i,k}^{\eta}\right) (d_{i,k})^2 + R \qquad (1)$$

with the probabilistic fuzzy assignments $u_{i,k} \in [0,1]$, the possibilistic typicality assignments $t_{i,k} \in [0,1]$ and m and η as the respective fuzzifiers. The additive

term

$$R = \sum_{k=1}^{M} \left(\delta_k \sum_{i=1}^{N} (t_{i,k} - 1)^{\eta} \right)$$

play the role of a regularization term. The value γ balances the influence of the probabilistic ($\gamma = 1$, FCM) and the possibilistic ($\gamma = 0$, PCM) model. A suggested choice for the δ_i-values is

$$\delta_k = K \frac{\sum_{i=1}^{N} (u_{i,k})^{\eta} (d_{i,k})^2}{\sum_{i=1}^{N} (u_{i,k})^{\eta}} \tag{2}$$

with the $u_{i,k}$ obtained from a pure FCM ($\gamma = 1$) and $\eta = m$ [21]. The constant $K > 0$ is a free parameter commonly chosen as $K = 1$. The constraints

$$\sum_{k=1}^{M} u_{i,k} = 1 \tag{3}$$

for FCM and

$$\sum_{i=1}^{N} t_{i,k} = 1 \tag{4}$$

have to be fulfilled. The constraint (3) requires the fuzzy assignments to be probabilistic for each data point $\mathbf{v}_i \in V$, whereas eq. (4) reflects the condition that the cluster typicality has to be probabilistic. The crisp c-means model is obtained for the FCM in the limit $m \to 1$, which leads to the optimal solution with $u_{i,k} \in \{0, 1\}$ [7, 24]. Yet, the latter condition is sufficient because for that case the optimum solution automatically leads to $u_{i,k} \in \{0, 1\}$.

If the Euclidean distance is used, optimization of the cost function (1) yields the alternating updates

$$\mathbf{w}_k = \frac{\sum_{i=1}^{N} \left(a \cdot u_{i,k}^m + b \cdot t_{i,k}^{\eta} \right) \mathbf{v}_i}{\sum_{i=1}^{N} \left(a \cdot u_{i,k}^m + b \cdot t_{i,k}^{\eta} \right)} \tag{5}$$

and

$$u_{i,k} = \frac{1}{\sum_{l=1}^{M} \left(\frac{d_{i,k}}{d_{i,l}} \right)^{\frac{2}{m-1}}} \tag{6}$$

for the assignments $u_{i,k}$ of FCM (3) using the alternating batch mode update strategy known from FCM. The typicality values $t_{i,k}$ are modified according to

$$t_{i,k} = \frac{1}{1 + \left(\frac{(d_{i,k})^2}{\delta_k} \right)^{\frac{1}{\eta - 1}}} \tag{7}$$

taking the δ_i-values into account.

2.2 Incorporation of Neighborhood Cooperativeness to Prevent Local Optima

Neighborhood cooperativeness between prototypes is a biologically inspired strategy to avoid prototype based vector quantizers to get stuck in local optima. Two main principles are widely applied: First, one can associate an external topological structure to the prototypes as introduced for self-organizing maps (SOMs) [19]. This external structure frequently is chosen to be a regular grid rather than another structures like a tree, yet, other structures are admissible. Suppose, that the external grid A is equipped with the dissimilarity measure d_A. In case of a regular grid this could be the Euclidean distance in A whereas for general graphs the minimal path length could be applied. Then the neighborhood cooperativeness between the prototypes is installed using a neighborhood function defined on A

$$h_\sigma^{SOM}(k,l) = c_\sigma \cdot \exp\left(-\frac{(d_A(k,l))^2}{2\sigma^2}\right) \tag{8}$$

with neighborhood range σ and the constraint $\sum_l h_\sigma^{SOM}(k,l) = 1$ ensured by the constant c_σ. Thus the neighborhood range induces a range of interactions also implicit in the data space determined by the location of the prototypes.

The alternative to the apriori fixed external neighborhood structure A would be a dynamic neighborhood determined by the actual distribution of the prototypes as suggested for the neural gas (NG) vector quantizer [25]. In NG the neighborhood between prototypes for a given data vector $\mathbf{v}_i \in V$ is based on the winning rank of each prototype $\mathbf{w}_k$

$$rk_k(\mathbf{v}_i, W) = \sum_{l=1}^{N} \Theta\left(d(\mathbf{v}_i, \mathbf{w}_k) - d(\mathbf{v}_i, \mathbf{w}_l)\right) \tag{9}$$

where

$$\Theta(x) = \begin{cases} 0 & if \quad x \leq 0 \\ 1 & else \end{cases} \tag{10}$$

is the Heaviside function [25]. The NG neighborhood function includes the ranks according to

$$\hat{h}_\sigma^{NG}(k|\mathbf{v}) = c_\sigma^{NG} \cdot \exp\left(-\frac{(rk_k(\mathbf{v}, W))^2}{2\sigma^2}\right) \tag{11}$$

with neighborhood range σ. This definition allows the declaration of a gradual

neighborhood relation between prototypes $\mathbf{w}_k$ and $\mathbf{w}_l$ by

$$h_\sigma^{NG}(k,l) = c_\sigma^{NG} \cdot \exp\left(-\frac{\left(rk_k(\mathbf{w}_l, W)\right)^2}{2\sigma^2}\right) \tag{12}$$

for a given neighborhood range σ. As before, the constraint $\sum_l h_\sigma^{NG}(k,l) = 1$ is ensured by a constant c_σ^{NG}.

Both, the external SOM as well as the dynamic NG neighborhood cooperativeness, induce local errors for a given $\mathbf{v}_i$ which are

$$lc_\sigma^{SOM}(i,k) = \sum_{l=1}^{M} h_\sigma^{SOM}(k,l) \cdot (d_{i,l})^2 \tag{13}$$

and

$$lc_\sigma^{NG}(i,k) = \sum_{l=1}^{M} h_\sigma^{NG}(k,l) \cdot (d_{i,l})^2, \tag{14}$$

respectively. These local errors can be used also for fuzzy vector quantization models instead of the quadratic distance [38, 39]. Replacing $(d_{i,k})^2$ in (1), the respective updates for optimization of the cost function (1) are obtained as

$$\mathbf{w}_k = \frac{\sum_{i=1}^{N}\sum_{l=1}^{M}\left(a \cdot u_{i,l}^m + b \cdot t_{i,l}^\eta\right) \cdot h_\sigma^{NG/SOM}(k,l) \cdot \mathbf{v}_i}{\sum_{i=1}^{N}\sum_{l=1}^{M}\left(a \cdot u_{i,l}^m + b \cdot t_{i,l}^\eta\right) \cdot h_\sigma^{NG/SOM}(k,l)} \tag{15}$$

if the Euclidean distance is used inside the local errors (13,14). We refer to these algorithms as Fuzzy-SOM (FSOM, [4, 3, 5, 27, 28, 37]) and Fuzzy-NG (FNG, [38]). The adaptation of the fuzzy assignments $u_{i,l}^m$ and the typicality assignments $t_{i,l}^m$ in the resulting FSOM/FNG are analogously to those in (6) and (7), respectively. Yet, now the dissimilarity measure $(d_{i,k})^2$ is replaced by the local costs $lc_\sigma^{NG}(i,k)$ and $lc_\sigma^{SOM}(i,k)$ accordingly:

$$u_{i,k} = \frac{1}{\sum_{l=1}^{M}\left(\frac{lc_\sigma^{NG/SOM}(i,k)}{lc_\sigma^{NG/SOM}(i,l)}\right)^{\frac{1}{m-1}}} \tag{16}$$

and

$$t_{i,k} = \frac{1}{1 + \left(\frac{\sum_{l=1}^{M} h_\sigma^{NG/SOM}(k,l) \cdot \left(d_{i,l}\right)^2}{\delta_k}\right)^{\frac{1}{\eta-1}}}. \tag{17}$$

For convergence details we refer to [4, 3, 5, 27, 28, 37] for FSOM and to [38, 9] for FNG.

3 Vector Quantizers for Classification Using Kernel Distances and Class Border Sensitive Learning

3.1 Utilization of Unsupervised Vector Quantizers for Classification Tasks and Class Border Sensitive Learning

Classification differs from unsupervised vector quantization in that each data point $\mathbf{v}_i \in V$ belongs to a certain class $c_i \in \{1, \ldots, C\}$. Different vector quantization schemes specifically are designed to deal with those problems. Prominent such models are the family of learning vector quantizers (LVQs, [19]) or generalizations thereof [29, 33, 32] as well as support vector machines (SVMs, [30]). These algorithms have in common that the prototypes $\mathbf{w}_k \in W = \{\mathbf{w}_j\}_{j=1}^{M} \subset \mathbb{R}^n$ are now responsible for the classes according to their class label $y_k \in \{1, \ldots, C\}$. Semi-supervised algorithms like the Fuzzy Labeled SOM/NG (FLSOM/NG,[44]) or the recently developed Fuzzy Supervised SOM/NG (FSSOM/NG, [16, 17]) assign fuzzy class labels to the prototypes, which are also adapted during the learning process. Yet, this adaptation is not independent from the prototype adaptation and the prototype adjustment is also influenced by the actual state of the labels. Moreover, at the end of the learning process the prototypes are class typical representatives.

Another way utilizing unsupervised vector quantizers is to take several vector quantizer networks, each of them responsible for one class. However, these networks should not simply act independently from each other. An information transfer between them is mandatory. One model is the Supervised Neural Gas (SNG,[11]) as a generalization of the LVQ2.1 algorithm. In SNG, on the one hand side, neighborhood cooperativeness between prototypes for the same class is installed for attraction forces according to NG. On the other hand, the repulsing forces are also modified according to the neighborhood relation between the prototypes of the incorrect classes for a given input. Again, in SNG the prototypes are class typical.

The combination of several FCM networks for data preprocessing in support vector machine learning in a two-class-problem is discussed in [36, 47]. The information transfer between the different FCM networks is realized by an *additional attraction force* for the best matching prototypes of both the correct and the incorrect class. Formally, this model can be expressed by the following cost function

$$E_{BS-FCM}\left(\mathbf{U},V\right)=\sum_{i=1}^{N_1}\sum_{k=1}^{M_1}u_{i,k}^{m}\left(1\right)\left(d_{i,k}\right)^2+\sum_{i=1}^{N_2}\sum_{k=1}^{M_2}u_{i,k}^{m}\left(2\right)\left(d_{i,k}\right)^2+F_{BS-FCM}\left(W,V\right) \tag{18}$$

with N_l, M_l, and $u_{i,k}^{m}\left(l\right)$ denoting the number of data samples in each subset V_l of V, the number of prototypes responsible for each data subset, and the fuzzy assignments according to the both classes. The attraction force term $F_{BS-FCM}\left(W,V\right)$ is

$$F_{BS-FCM}\left(W,V\right)=\sum_{i=1}^{N}d\left(\mathbf{w}_{s_1(i)},\mathbf{w}_{s_2(i)}\right) \tag{19}$$

where $s_l\left(i\right)$ denotes the closest prototype responding to class l for given input $\mathbf{v}_i$. This term enforces the prototypes to move to the class borders as known from SVMs and, therefore, they are not longer class typical. Thus we obtain a border sensitive supervised FCM model (BS-FCM) for the two-class-problem as proposed in [36, 47].

Obviously, this BS-FCM method can be generalized in several ways: First, it is immediately applicable to the PFCM cost function $E_{PFCM}\left(\mathbf{U},V,W,\delta,\gamma\right)$ in (1). Second, the generalization to more than two classes is possible by redefining the attraction force (19) as

$$F\left(W,V\right)=\sum_{i=1}^{N}d\left(\mathbf{w}_{s^+(i)},\mathbf{w}_{s^-(i)}\right) \tag{20}$$

where $s^+\left(i\right)$ and $s^-\left(i\right)$ are determining the closest prototype of the correct class and the closest prototype of all incorrect classes for a given data vector $\mathbf{v}_i$. Thus, we obtain the border sensitive PFCM (BS-PFCM) with the cost function

$$E_{BS-PFCM}=\sum_{l=1}^{C}\left[\sum_{k=1}^{M_l}\sum_{i=1}^{N_l}\left(\gamma\cdot u_{i,k}^{m}\left(l\right)+\left(1-\gamma\right)\cdot t_{i,k}^{\eta}\left(l\right)\right)\left(d_{i,k}\right)^2+R_l\right]+F\left(W,V\right) \tag{21}$$

to be minimized, where

$$R_l=\sum_{k=1}^{M_l}\left(\delta_k\sum_{i=1}^{N_l}\left(t_{i,k}\left(l\right)-1\right)^{\eta}\right)$$

is the regularization term for the lth class model.

3.2 Neighborhood Cooperativeness in Border Sensitive Learning

Now, we act on the suggestion of neighborhood cooperativeness as a method to improve the convergence. As mentioned before, neighborhood cooperativeness between the prototypes within a PFCM instance leads to a FNG-variant or FSOM-variant depending on the applied crisp vector quantizer. This principle can be transferred to neighborhood cooperativness between the prototypes of different instances with respect to class border sensitive learning. We refer to the resulting models as *border sensitive FSOM* (BS-FSOM) and *border sensitive FNG* (BS-FNG).

The incorporation of the local costs in the first term of (21) is straightforward handling each sub-network as a single FSOM or FNG, respectively. More attention has to be given to an appropriate redefinition of the attraction force $F(W,V)$ from (20): Let W_i^- be the set of all prototypes which are of different classes than the class c_i for a given data point $\mathbf{v}_i$ and

$$h_{\sigma_-}^{NG}\left(k,l,W^-\right) = c_{\sigma_-}^{NG} \cdot \exp\left(-\frac{\left(rk_k\left(\mathbf{w}_l,W^-\right)\right)^2}{2\sigma_-^2}\right) \tag{22}$$

is a NG-like neighborhood function according to (12) but restricted to W^- with neighborhood range σ_-. Then the new *neighborhood-attentive* attraction force (NAAF) is defined as

$$F_{neigh}(W,V) = \sum_{i=1}^{N} \sum_{k=1 \wedge \mathbf{w}_k \in W^-}^{M} h_{\sigma_-}^{NG}\left(k, s^+(i), W^-\right) d\left(\mathbf{w}_{s^+(i)}, \mathbf{w}_k\right) \tag{23}$$

which reduces to $F_{neigh}(W,V)$ in (20) of the BS-FCM for $\sigma_- \to 0$. The force $F_{neigh}(W,V)$ again compels the prototypes to move to the class borders. However, the neighborhood cooperativeness speeds up this process scaled by neighborhood range σ. Thereby, the responsibilities of the prototypes for the different class borders are not predetermined,rather they are a result of a self-organizing process, which provides a great robustness and stability. As previously, border sensitive learning in FSOM and FNG is obtained replacing the quadratic distances $(d_{i,k})^2$ by the local costs $lc_\sigma^{SOM/NG}(i,k)$ from (13,14) in BS-PFCM (21) together with the NAAF $F_{neigh}(W,V)$.

3.3 Kernel Distances and their Use in Border Sensitive Vector Quantization Classification

All the above algorithms are based on the evaluation of the dissimilarity between data and prototypes, frequently chosen as the (quadratic) Euclidean metric. During the last years, other dissimilarity measures are investigated to improve the classification and vector quantization abilities for different tasks. Among them adaptive quadratic forms [31], the scaled Euclidean metric [12], or functional norms [15, 22, 23] like divergences [40] or Sobolev-norms [43] became popular. Another strategy is considered in Support Vector Machines (SVMs) [30, 34].

3.3.1 Kernel distances

We consider the data space V. In SVMs, the data $\mathbf{v} \in V$ are implicitly mapped into a high- maybe infinite-dimensional function Hilbert space $\mathcal{H}$ with the metric $d_{\mathcal{H}}$, which offers a rich topological structure, such that classes become easily separable in that space [30, 35]. The implicit mapping $\Phi : V \to \mathcal{I}_{\kappa_\oplus} \subseteq \mathcal{H}$ is determined by a kernel $\kappa_\Phi(\mathbf{v}, \mathbf{w})$ defining an inner product in $\mathcal{H}$ but being evaluated in the data space. If the mapping Φ is universal, the span $\mathcal{I}_{\kappa_\oplus}$ of the image $\Phi(V)$ forms a subspace of $\mathcal{H}$ with the kernel induced metric

$$d_{\kappa_\Phi}(\mathbf{v}, \mathbf{w}) = \sqrt{\kappa_\Phi(\mathbf{v}, \mathbf{v}) + 2\kappa_\Phi(\mathbf{v}, \mathbf{w}) + \kappa_\Phi(\mathbf{w}, \mathbf{w})}, \tag{24}$$

which coincide with the Hilbert space metric $d_{\mathcal{H}}$ [35]. The disadvantage of SVMs is that the function Hilbert space is not longer intuitive like the data space. However, it turns out that for *differentiable universal* kernels an identical mapping $\Psi : (V, d_V) \longrightarrow \mathcal{V}$ can be applied to the data, such that both prototypes and data remain as they are but get equipped with the kernel metric $d_{\kappa_\Phi} = d_{\mathcal{H}}$ in the mapping space $\mathcal{V} = (V, d_{\mathcal{H}})$. This new mapping space $\mathcal{V}$ is isomorphic to the mapping space I_{κ_Φ} of the mapping Φ [41], see Fig. 3.3.1.

3.3.2 Kernel Distances in BS-FCM

Obviously, kernel distances can be applied in FCM and PFCM as well as in BS-FNG/FSOM replacing the distances $d_{i,k}$ by $d_{\kappa_\Phi}(\mathbf{v}_i, \mathbf{v}_j)$ as given above in (24). In the border sensitive models BS-FNG/FSOM, the prototypes are positioned near the class borders in the mapping space $\mathcal{V}$ according to the previously explained strategy of border sensitive learning. Because $\mathcal{V}$ is isomorphic to the Hilbert space $\mathcal{H}$ of SVMs, we can immediately interpret the prototypes as approximations of

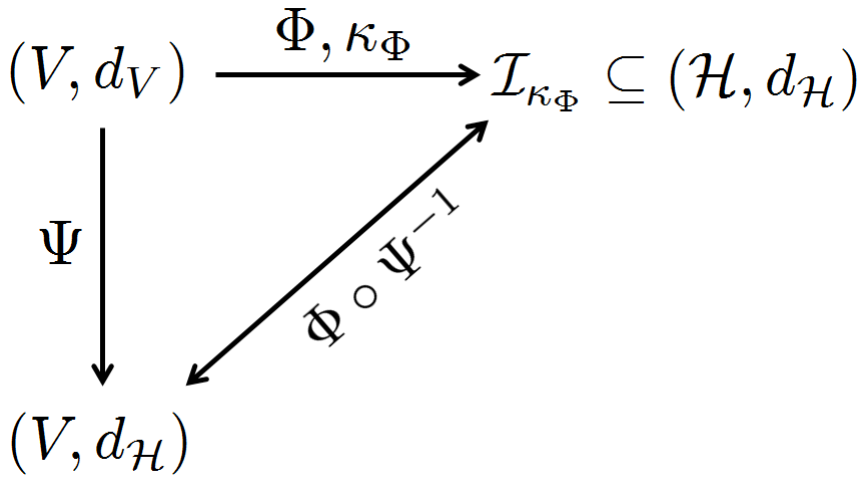


Figure 1: Visualization of the relations between the data space V and the mapping spaces $\mathcal{I}_{\kappa_\Phi}$ and $\mathcal{V}$ for the mappings Φ and Ψ, respectively: For universal kernels κ_Φ the metric spaces $\mathcal{V} = (V, d_\mathcal{H})$ and $\left(\mathcal{I}_{\kappa_\oplus}, d_\mathcal{H}\right)$ are topologically equivalent and isometric by means of the continuous bijective mapping $\Phi \circ \Psi^{-1}$. (Figure taken from [41]).

support vectors for that case. We refer to this model as *border sensitive kernel* FNG/FSOM (BS-KFNG/SOM). In contrast to SVMs, the complexity of the model is fixed in advance according to the number of prototypes used. In this way, it is not longer only a preprocessing scheme like BS-FCM proposed in [36, 47], but rather a standalone fuzzy classification model based on kernel distances. Yet, the resulting updates may become intractable as explicite rules.

So far, we adressed the problem of fuzzy classification following the idea of utilizing unsupervised models for this task. However, other prototype based fuzzy classification algorithms could be treated analogously by an additive border sensitivity force like (23), if the respective model posseses a cost function. As a prominent example we mention the Fuzzy Soft Nearest Prototype Classifier (FSNPC) proposed in [46].

4 Conclusion

In this paper we investigate ideas how to generate class border sensitive fuzzy classification schemes based on interacting unsupervised fuzzy vector quantization models. We give the theoretical background forthese models based on the family

of fuzzy-c-means quantizers. In particular, we focus on a dynamic neighborhood between the unsupervised models responsible for each class based on a neighborhood relation scheme known from neural gas. Although we considered only fuzzy vector quantizers, the extensions to other schemes like the common NG or SOM is straight forward.

References

[1] J. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum, New York, 1981.

[2] J. Bezdek, R. Hathaway, and M. Windham. Numerical comparison of RFCM and AP algorithms for clustering relational data. *Pattern recognition*, 24:783–791, 1991.

[3] J. C. Bezdek and N. R. Pal. A note on self-organizing semantic maps. *IEEE Transactions on Neural Networks*, 6(5):1029–1036, 1995.

[4] J. C. Bezdek and N. R. Pal. Two soft relatives of learning vector quantization. *Neural Networks*, 8(5):729–743, 1995.

[5] J. C. Bezdek, E. C. K. Tsao, and N. R. Pal. Fuzzy Kohonen clustering networks. In *Proc. IEEE International Conference on Fuzzy Systems*, pages 1035–1043, Piscataway, NJ, 1992. IEEE Service Center.

[6] C. Brüß, F. Bollenbeck, F.-M. Schleif, W. Weschke, T. Villmann, and U. Seiffert. Fuzzy image segmentation with fuzzy labeled neural gas. In M. Verleysen, editor, *Proc. Of European Symposium on Artificial Neural Networks (ESANN'2006)*, pages 563–568, Brussels, Belgium, 2006. d-side publications.

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

[8] J. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3:32–57, 1973.

[9] T. Geweniger, M. Kästner, M. Lange, and T. Villmann. Modified CONN-index for the evaluation of fuzzy clusterings. In M. Verleysen, editor, *Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2012)*, pages 465–470, Louvain-La-Neuve, Belgium, 2012. i6doc.com.

[10] T. Geweniger, D. Zühlke, B. Hammer, and T. Villmann. Median fuzzy c-means for clustering dissimilarity data. *Neurocomputing*, 73(7–9):1109–1116, 2010.

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

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

[13] H. Ichihashi and K. Honda. Application of kernel trick to fuzzy c-means with regularization by K-L information. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 8(6):566–572, 2004.

[14] R. Inokuchi and S. Miyamoto. Fuzzy c-means algorithms using Kullback-Leibler divergence and Hellinger distance based on multinomial manifold. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 12(5):443–447, 2008.

[15] M. Kästner, B. Hammer, M. Biehl, and T. Villmann. Functional relevance learning in generalized learning vector quantization. *Neurocomputing*, 90(9):85–95, 2012.

[16] M. Kästner, W. Hermann, and T. Villmann. Integration of structural expert knowledge about classes for classification using the fuzzy supervised neural gas. In M. Verleysen, editor, *Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2012)*, pages 209–214, Louvain-La-Neuve, Belgium, 2012. i6doc.com.

[17] M. Kästner, M. Lange, and T. Villmann. Fuzzy supervised self-organizing map for semi-supervised vector quantization. In L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. Zadeh, and J. Zurada, editors, *Artificial Intelligence and Soft Computing - Proc. the International Conference ICAISC, Zakopane*, volume 1 of *LNAI 7267*, pages 256–265, Berlin Heidelberg, 2012. Springer.

[18] M. Kästner and T. Villmann. Fuzzy supervised neural gas for semi-supervised vector quantization – theoretical aspects. *Machine Learning Reports*, 5(MLR-02-2011):1–16, 2011. ISSN:1865-3960, http://www.techfak.uni-bielefeld.de/~fschleif/mlr/mlr_02_2011.pdf.

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

[20] R. Krishnapuram and J. Keller. A possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems*, 1(4):98–110, 1993.

[21] R. Krishnapuram and J. Keller. The possibilistic c-means algorithm: insights and recommendations. *IEEE Transactions on Fuzzy Systems*, 4(3):385–393, 1996.

[22] 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.

[23] J. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. Information Sciences and Statistics. Springer Science+Business Media, New York, 2007.

[24] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28:84–95, 1980.

[25] 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.

[26] N. Pal, K. Pal, J. Keller, and J. Bezdek. A possibilistic fuzzy c-means clustering algorithm. *IEEE Transactions on Fuzzy Systems*, 13(4):517–530, 2005.

[27] N. R. Pal, J. C. Bezdek, and E. C. K. Tsao. Generalized clustering networks and Kohonen's self-organizing scheme. *IEEE Transactions on Neural Networks*, 4(4):549–557, 1993.

[28] N. R. Pal, J. C. Bezdek, and E. C. K. Tsao. Errata to Generalized clustering networks and Kohonen's self-organizing scheme. *IEEE Transactions on Neural Networks*, 6(2):521–521, March 1995.

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

[31] P. Schneider, B. Hammer, and M. Biehl. Adaptive relevance matrices in learning vector quantization. *Neural Computation*, 21:3532–3561, 2009.

[32] S. Seo, M. Bode, and K. Obermayer. Soft nearest prototype classification. *IEEE Transaction on Neural Networks*, 14:390–398, 2003.

[33] S. Seo and K. Obermayer. Soft learning vector quantization. *Neural Computation*, 15:1589–1604, 2003.

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

[35] I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2001.

[36] S. Tian, S. Mu, and C. Yin. Cooperative clustering for training SVMs. In J. Wang, Z. Yi, J. Zurada, B.-L. Lu, and H. Yin, editors, *Advances in Neural Networks - Third International Symposium on Neural Networks (ISNN 2006), Chengdu, China*, volume 3971 of *LNCS*, pages 962–967. Springer, 2006.

[37] E. Tsao, J. Bezdek, and N. Pal. Fuzzy Kohonen clustering networks. *Pattern Recognition*, 27(5):757–764, 1994.

[38] T. Villmann, T. Geweniger, M. Kästner, and M. Lange. Theory of fuzzy neural gas for unsupervised vector quantization. *Machine Learning Reports*, 5(MLR-06-2011):27–46, 2011. ISSN:1865-3960, http://www.techfak.uni-bielefeld.de/~fschleif/mlr/mlr_06_2011.pdf.

[39] T. Villmann, T. Geweniger, M. Kästner, and M. Lange. Fuzzy neural gas for unsupervised vector quantization. In L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. Zadeh, and J. Zurada, editors, *Artificial Intelligence and Soft Computing - Proc. the International Conference ICAISC, Zakopane*, volume 1 of *LNAI 7267*, pages 350–358, Berlin Heidelberg, 2012. Springer.

[40] T. Villmann and S. Haase. Divergence based vector quantization. *Neural Computation*, 23(5):1343–1392, 2011.

[41] T. Villmann and S. Haase. A note on gradient based learning in vector quantization using differentiable kernels for Hilbert and Banach spaces. *Machine Learning Reports*, 6(MLR-02-2012):1–29, 2012. ISSN:1865-3960, http://www.techfak.uni-bielefeld.de/~fschleif/mlr/mlr_02_2012.pdf.

[42] T. 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.

[43] T. Villmann and B. Hammer. Functional principal component learning using Ojas method and Sobolev norms. In J. Principe and R. Miikkulainen, editors, *Advances in Self-Organizing Maps - Proceeding of the Workshop on Self-Oranizing Maps (WSOM)*, pages 325–333. Springer, 2009.

[44] T. Villmann, B. Hammer, F.-M. Schleif, T. Geweniger, and W. Herrmann. Fuzzy classification by fuzzy labeled neural gas. *Neural Networks*, 19:772–779, 2006.

[45] T. Villmann, E. Merényi, and W. Farrand. Unmixing hyperspectral images with fuzzy supervised self-organizing maps. In M. Verleysen, editor, *Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2012)*, pages 185–190, Louvain-La-Neuve, Belgium, 2012. i6doc.com.

[46] T. Villmann, F.-M. Schleif, and B. Hammer. Prototype-based fuzzy classification with local relevance for proteomics. *Neurocomputing*, 69(16–18):2425–2428, October 2006.

[47] C. Yin, S. Mu, and S. Tian. Using cooperative clustering to solve multiclass problems. In Y. Wang and T. Li, editors, *Foundation of Intelligent Systems - Proc. of the Sixth International Conference on Intelligent Systems and Knowledge Engineering (ISKE 2011), Shanghei, China*, volume 122 of *Advances in Intelligent and Soft Computing*, pages 327–334. Springer, 2012.

Class Border Sensitive Generalized Learning Vector Quantization

- *An Alternative to Support Vector Machines* -

M. Kästner[1], M. Riedel[1], M. Strickert[2], and T. Villmann[1*]

[1]Computational Intelligence Group, University of Applied Sciences Mittweida, Technikumplatz 17, 09648 Mittweida, Germany

[2]Computational Intelligence Group, Philipps-University Marburg, Hans-Meerwein-Straße 6, 35032 Marburg , Germany

Abstract

Prototype based classification models like learning vector quantization or support vector machines (SVMs) take into account the class distribution in the data space for representation and class discrimination. While support vectors indicate the class borders, prototypes in learning vector quantzation roughly represent the class centers. Depending on the task, both strategies provide advantages or disadvantages. Generalized learning vector quantization (GLVQ) proposed by Sato&Yamada offers, besides the aspect that is here learning is based on a cost function, additional possibilities to control the learning processs. In this paper we emphasize the ability to establish a class border sensitive learning of the prototypes by means of appropriate choices of the parameter in cost function of this model. Alternatively, we discuss additive penalty functions to force the prototypes to ferret out the class borders. In this way, the application range of GLVQ is extended also covering those areas, which are only in the focus of SVMs so far because of the required precise detection of class borders.

*corresponding author, *email: thomas.villmann@hs-mittweida.de*

1 Introduction

One of the most promising concepts in classification by prototype based models is the family of KOHONEN'S Learning Vector Quantizers (LVQ,[7]). Although only heuristically motivated these algorithms frequently show great performance also in comparison with sophisticated approaches like support vector machines (SVM, [14, 22]) or multilayer perceptrons [6]. The main idea in LVQ is to distribute prototypes as class representatives in the data space. During the learning the prototypes are moved toward the randomly presented data points or pushed away depending on their predefined class responsibilities to increase the classification accuracy. The classification decision for a given data vector is made considering the dissimilarity of the vector to the model prototypes. Frequently, the dissimilarity is based on a distance measure like the Euclidean distance. Starting from the basic heuristic LVQ scheme many algorithms were established keeping these principles but bearing down the pure heuristic learning dynamic. In particular, those algorithms approximate the classification accuracy by cost functions to be minimized throughout the learning phase. Prominent examples of such approaches are the generalized LVQ (GLVQ,[13]), Soft Learning Vector Quantization (SLVQ) and Soft Nearest Prototype Classification (SNPC) [20, 19]. All these models have in common that the prototype are located inside the borders of class distribution. In contrast, SVMs determine data vectors defining the class borders such that these serve as prototypes denoted as support vectors in this context [3, 14, 22]. Here, the classification learning is performed after implicit mapping of the data into a high-, and potentially infinite-, dimensional Hilbert space by kernel mapping using the theory of reproducing kernel Hilbert spaces (RKHS, [23]). This high-dimensional mapping offers great flexibility in learning, which provides together with the class border sensitive support vector principle high classification accuracy abilities.

Recent investigations have shown that kernel based learning is also possible for LVQ methods using universal differentiable kernels [28]. In this paper we focus on class border sensitive learning in GLVQ. For this purpose we consider two different strategies: First we consider the influence of the activation function of the classifier function in GLVQ as suggested by [29]. Second, we introduce an additive term for the cost function in GLVQ forcing the prototypes to move to the class borders. We show that both strategies lead to the desired class border sensitive prototype adjustment.

The paper is organized as follows: After a brief review of GLVQ and variants thereof we provide both strategies for class border sensitive learning in GLVQ. Il-

lustrating examples show the abilities of the approaches. Some concluding remarks complete the paper.

2 Generalized Learning Vector Quantization (GLVQ)

In this section we briefly revisit the GLVQ and some of it variants neither claiming completeness nor investigating all details. We only focus on these properties relevant for the topic considered here.

2.1 The basic GLVQ

Basic GLVQ was published by SATO & YAMADA in [13]. The aim was to keep the basic principle of attraction and repulsion in prototype based classification learning in LVQ but vanquishing the problem of the adaptation heuristic in standard LVQ as suggested by KOHONEN [7]. Precisely, given a set $V \subseteq \mathbb{R}^D$ of data vectors $\mathbf{v}$ with class labels $x_{\mathbf{v}} \in \mathcal{C} = \{1, 2, \ldots C\}$ and N prototypes $\mathbf{w}_j \in W \subset \mathbb{R}^D$ with class labels $y_j \in \mathcal{C}$ $(j = 1, \ldots, N)$, the GLVQ introduces a cost function

$$E_{GLVQ}(W) = \sum_{\mathbf{v} \in V} f(\mu(\mathbf{v})) \tag{1}$$

where the dependence on W is implicitly given by the classifier funvtion

$$\mu(\mathbf{v}) = \frac{d^+(\mathbf{v}) - d^-(\mathbf{v})}{d^+(\mathbf{v}) + d^-(\mathbf{v})} \tag{2}$$

is the *classifier function*, via $d^+(\mathbf{v}) = d(\mathbf{v}, \mathbf{w}^+)$ denoting the distance between the data vector $\mathbf{v}$ and the closest prototype $\mathbf{w}^+$ with the same class label $y^+ = x_{\mathbf{v}}$, and $d^-(\mathbf{v}) = d(\mathbf{v}, \mathbf{w}^-)$ is the distance to the best matching prototype $\mathbf{w}^-$ with a class label y^- different from $x_{\mathbf{v}}$. Frequently, the squared Euclidean distance is used. We remark that $\mu(\mathbf{v}) \in [-1, 1]$ holds. The *transfer function* f is the monotonically increasing and frequently taken as a sigmoid function or the identity function $f(t) = t$. If we take the logistic function

$$f_\theta(\mu) = \frac{1}{1 + \exp\left(\frac{-\mu}{2\theta^2}\right)} \tag{3}$$

with $0 < f_\theta(\mu) < 1$, see Fig. 1.

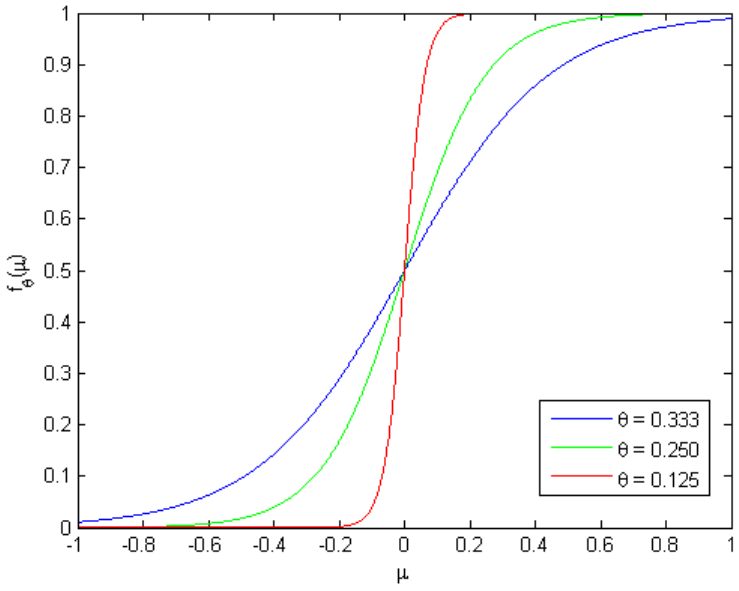


Figure 1: Visualization of the sigmoid transfer function $f_\theta(\mu)$ from (3) for different parameter values θ.

For $\theta \to 0$ the logistic function $f_\theta(\mu)$ converges to the Heaviside function

$$H(x) = \begin{cases} 0 & if \quad x \leq 0 \\ 1 & else \end{cases} . \tag{4}$$

In this limit the cost functions $E_{GLVQ}(W)$ counts the misclassifications.

Learning takes place as stochastic gradient descent on $E_{GLVQ}(W)$. In particular we have

$$\triangle \mathbf{w}^+ \sim \xi^+(\mathbf{v}) \cdot \frac{\partial d^+(\mathbf{v})}{\partial \mathbf{w}^+} \text{ and } \triangle \mathbf{w}^- \sim \xi^-(\mathbf{v}) \cdot \frac{\partial d^-(\mathbf{v})}{\partial \mathbf{w}^-} \tag{5}$$

with the scaling factors

$$\xi^+(\mathbf{v}) = f'(\mu(\mathbf{v})) \cdot \frac{2 \cdot d^-(\mathbf{v})}{(d^+(\mathbf{v}) + d^-(\mathbf{v}))^2} \tag{6}$$

and

$$\xi^-(\mathbf{v}) = -f'(\mu(\mathbf{v})) \cdot \frac{2 \cdot d^+(\mathbf{v})}{(d^+(\mathbf{v}) + d^-(\mathbf{v}))^2}. \tag{7}$$

For the quadratic Euclidean metric we simply have the derivatives

$$\frac{\partial d^\pm(\mathbf{v})}{\partial \mathbf{w}^\pm} = -2\left(\mathbf{v} - \mathbf{w}^\pm\right)$$

realizing a vector shift of the prototypes in the data space.

Remark 2.1 *Although one could think about degenerated data distributions $P(V)$ in V we always assume that the set W of all prototypes is separable, i.e. $d(\mathbf{w}_j, \mathbf{w}_l) > \varepsilon_V$ for $j \neq l$ with a certain data dependent $\varepsilon_V > 0$.*

This remark ensures a non-zero denominator in the classifier function (2) and respective derivatives.

2.2 GLVQ and non-Euclidean distances

As mentioned above, frequently the (squared) Euclidean distance (metric) is used in GLVQ. Yet, depending on the classification problem other dissimilarity measures may be more appropriate [5]. For GLVQ the dissimilarity measure $d(\mathbf{v}, \mathbf{w})$ has not necessarily to be a mathematical distance but assumed to be at least a dissimilarity measure [11], which is differentiable in the second argument.[1] Thus the scaled (squared) Euclidean metric

$$d_{\mathbf{\Lambda}}(\mathbf{v}, \mathbf{w}) = (\mathbf{v}, \mathbf{w})^{\top} \mathbf{\Lambda} (\mathbf{v}, \mathbf{w}) \tag{8}$$

with the diagonal matrix $\Lambda = diag(\lambda_1, \ldots, \lambda_D)$ and $\lambda_i \geq 0$ was considered in [5]. Here, the diagonal values λ_i are also adjusted via a gradient descent scheme. Generalizations of this approach take the positive semi-definite matrix $\mathbf{\Lambda}$ as a matrix product $\mathbf{\Lambda} = \Omega^{\top}\Omega$ with arbitrary matrices $\Omega \in \mathbb{R}^{m \times D}$ to be optimized during the training [2, 16, 17, 18]. In case of functional data, i.e. the data vectors are discrete representations of positive functions, divergences are proposed as appropriate dissimilarities [10, 27].

Recent considerations deal with kernel distances

$$d_{\kappa_\Phi}(\mathbf{v}, \mathbf{w}) = \sqrt{\kappa_\Phi(\mathbf{v}, \mathbf{v}) + 2\kappa_\Phi(\mathbf{v}, \mathbf{w}) + \kappa_\Phi(\mathbf{w}, \mathbf{w})}, \tag{9}$$

as dissimilarity measure [28]. In this distance $\kappa_\Phi(\mathbf{v}, \mathbf{w})$ is an universal differentiable kernel [23]. The kernel $\kappa_\Phi(\mathbf{v}, \mathbf{w})$ implicitly defines a generally non-linear mapping $\Phi : V \to \mathcal{I}_{\kappa_\oplus} \subseteq \mathcal{H}$ of the data and prototypes into a high- maybe infinite-dimensional function Hilbert space $\mathcal{H}$ with the metric $d_{\mathcal{H}}(\Phi(\mathbf{v}), \Phi(\mathbf{w})) = d_{\kappa_\Phi}(\mathbf{v}, \mathbf{w})$ [1, 9]. For universal kernels the image $\mathcal{I}_{\kappa_\oplus} = span(\Phi(V))$ forms a subspace of $\mathcal{H}$ [23]. For differentiable universal kernels we can define an accompanying

[1] The last weak assumption assumes that the dissimilarity measure is always used in this manner that the adaptive prototype is the second argument, as it is usual in the field.

transformation $\Psi : V \longrightarrow \mathcal{V}$, where in $\mathcal{V}$ the data are equipped with the kernel metric d_{κ_Φ}. The generally non-linear map Ψ is bijective iff Φ does, i.e. iff the kernel is universal [23]. It turns out that $\mathcal{V}$ is an isometric isomorphism to $\mathcal{I}_{\kappa_\oplus}$, and the differentiability of the kernel ensures the applicability of the stochastic gradient learning of GLVQ in $\mathcal{V}$ for the kernel distance [28]. Hence, the resulting kernel GLVQ (KGLVQ) is running in the new data space $\mathcal{V}$ which offers the same topological structure and richness as the image $\mathcal{I}_{\kappa_\oplus}$, which is used in SVMs as the underlying dual data structure. We denote this new data space as *kernelized data space*. However, as explained in the introduction, the prototypes in KGLVQ remain to be class typical and are not focusing to detect the class borders for class discrimination.

3 Class Border Sensitive Learning in GLVQ

As we have seen in the previous section, GLVQ can be proceeded using kernel distances while prototypes remain class typical. This might be a disadvantage if precise decisions are favored toward class typical prototypes with maybe slightly decreased accuracy. For this situation it would be desirable to have a GLVQ variant offering such an ability. In this section we provide two possibilities to do so in GLVQ. The first one uses parametrized sigmoid transfer functions f, where the parameter controls the class border sensitivity. The second approach applies an additive attraction force for prototypes with different class responsibilities.

3.1 Class Border Sensitive Learning by Parametrized Transfer Functions in GLVQ

Following the explanations in [24, 29], we investigate in this subsection the influence of an appropriate chosen parametrized transfer function f to be applied in the cost function (1) of GLVQ. For the considerations here the logistic function (3) is used. It is well-known that the derivative $f'_\theta(\mu(\mathbf{v}))$ of the logistic function can be expressed as

$$f'_\theta(\mu(\mathbf{v})) = \frac{f_\theta(\mu(\mathbf{v}))}{2\theta^2} \cdot (1 - f_\theta(\mu(\mathbf{v}))), \tag{10}$$

which appears in the scaling factors in $\xi^\pm$ (6) and (7) for the winning prototypes $\mathbf{w}^\pm$. Looking at these derivatives (see Fig. 2) we observe that a significant prototype update only takes place for a small range of the classifier values μ in (2) depending on the parameter θ.

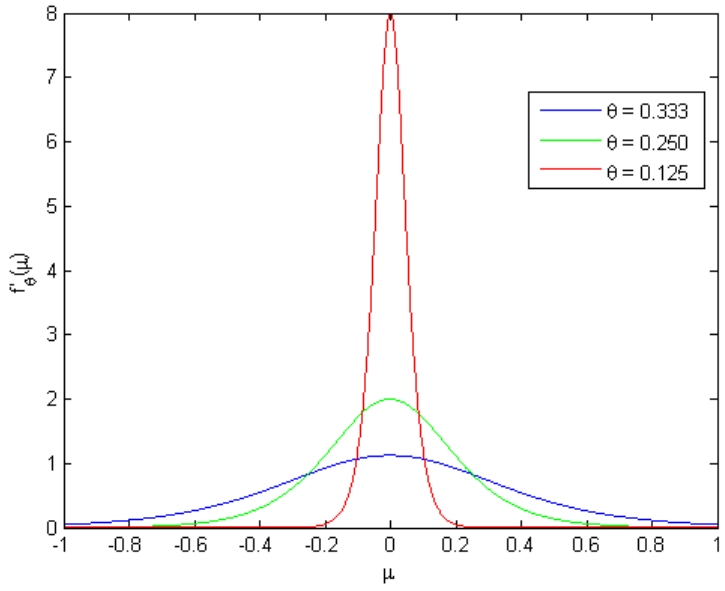


Figure 2: Visualization of the derivative sigmoid transfer function $f_\theta(\mu)$ from (3) for different parameter values θ.

Hence, we consider the set

$$\Xi = \left\{ \mathbf{v} \in V | \mu(\mathbf{v}) \in \left] -\frac{1-\mu_\theta}{1+\mu_\theta}, \frac{1-\mu_\theta}{1+\mu_\theta} \right[\right\}$$

with μ_θ chosen such that $f'_\theta(\mu) \approx 0$ is valid for $\mu \in \Xi$. Complementarily we define the *active set*

$$\hat{\Xi} = V \backslash \Xi \tag{11}$$

of the data contributing significantly to a prototype update, see Fig. 3. Obviously, the active set is distributed along the class decision bounderies, because only there $f'_\theta(\mu) \gg 0$ is valid. This corresponds to $\mu(\mathbf{v}) \approx 0$. Hence, this active set $\hat{\Xi}$ can be understood as another formulation of Kohnen's window rule in LVQ2.1

$$\min\left(\frac{d^+(\mathbf{v})}{d^-(\mathbf{v})}, \frac{d^-(\mathbf{v})}{d^+(\mathbf{v})}\right) \geq \frac{1-w}{1+w} \tag{12}$$

taking there $w = \mu_\theta$ [7, 29]. A similar rule was also obtained for SLVQ and SNPC [20, 19]. It was used to optimize learning in these algorithms in [21, 15]. The learning of the parameter θ in GLVQ was explicitly addressed in [29]. Optimization for accuracy improvement was discussed in [24].

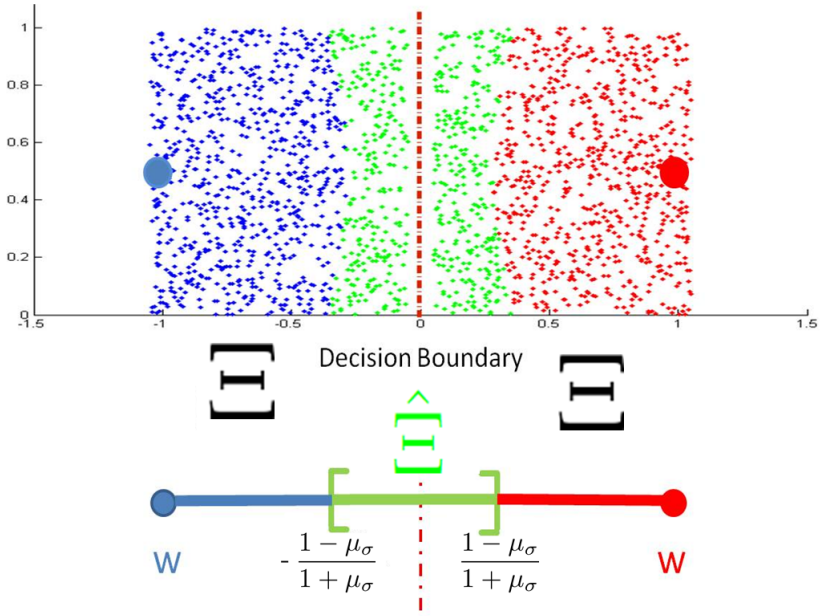


Figure 3: Visualization of the active set $\hat{\Xi}$ (green points) for a simple example.

Here we emphasize the aspect that the parameter θ allows a *control of the width of the active set* surrounding the class borders. Small θ-values define small stripes as active sets. In consequence, only these data contribute to the prototype updates. In other words, according to (11) the active set is crisp but the possibilities for control are smooth such that we could speak about *thresholded active sets*. Hence, border sensitive leads to prototype locations close to the class borders depending on the control parameter θ.

3.2 Border Sensitive Learning in GLVQ by a Penalty Function

Class border sensitivity learning by an additive penalty term was proposed for two-class-problems using two unsupervised fuzzy-c-means models in [30]. The generalization for more-class-problems and incorporation of neighborhood cooperativeness for convergence improvement is recently proposed in [4]. Here we adopt these ideas for class border sensitive learning in GLVQ (BS-GLVQ).

For this purpose we suggest a cost function $E_{BS-GLVQ}(W)$ as a convex sum

$$E_{BS-GLVQ}(W,\gamma) = (1-\gamma)\cdot E_{GLVQ}(W) + \gamma \cdot F_{neigh}(W,V) \tag{13}$$

with the new *neighborhood-attentive attraction force* (NAAF)

$$F_{neigh}\left(W,V\right)=\sum_{\mathbf{v}\in V}^{N}\sum_{k:\mathbf{w}_k\in W^-(\mathbf{v})}h_{\sigma}^{NG}\left(k,\mathbf{w}^+,W^-\left(\mathbf{v}\right)\right)d\left(\mathbf{w}^+,\mathbf{w}_k\right) \tag{14}$$

and the sensitivity control parameter $\gamma \in (0,1)$. The set $W^-(\mathbf{v}) \subset W$ is the set of all prototypes with incorrect class labels for a given data vector $\mathbf{v}$. The neighborhood function

$$h_{\sigma_-}^{NG}\left(k,\mathbf{w}^+,W^-\left(\mathbf{v}\right)\right)=c_{\sigma_-}^{NG}\cdot\exp\left(-\frac{\left(rk_k\left(\mathbf{w}^+,W^-\left(\mathbf{v}\right)\right)-1\right)^2}{2\sigma_-^2}\right) \tag{15}$$

defines a neighborhood of the prototypes in $W^-(\mathbf{v})$ with respect to the best matching correct prototype $\mathbf{w}^+$. Here $rk_k\left(\mathbf{w}^+,W^-\left(\mathbf{v}\right)\right)$ is the dissimilarity rank function of the prototypes $\mathbf{w}_k \in W^-(\mathbf{v})$ with respect to $\mathbf{w}^+$ defined as

$$rk_k\left(\mathbf{w}^+,W^-\left(\mathbf{v}\right)\right)=\sum_{\mathbf{w}_l\in W^-(\mathbf{v})}H\left(d\left(\mathbf{w}^+,\mathbf{w}_k\right)-d\left(\mathbf{w}^+,\mathbf{w}_l\right)\right) \tag{16}$$

with H being the Heaviside function (4). The neighborhood range is implicitly controlled by the parameter $\sigma_- > 0$. This kind of neighborhood function is known from *Neural Gas* (NG,[8]) with the modification used in Fuzzy-NG [26, 25].

Remark 3.1 *We emphasize at this point that the NAAF $F_{neigh}\left(W,V\right)$ depends on the dissimilarity measure used for GLVQ learning via (16). However, it is robust because only ranks of dissimilarities are involved.*

Because of the assumed separability of W (see Remark 2.1) the NAAF $F_{neigh}\left(W,V\right)$ is differentiable such that gradient based learning is possible. We have

$$\frac{\partial F_{neigh}\left(W,V\right)}{\partial\mathbf{w}_j}=h_{\sigma_-}^{NG}\left(j,\mathbf{w}^+,W^-\left(\mathbf{v}\right)\right)\cdot\frac{\partial d\left(\mathbf{w}^+,\mathbf{w}_j\right)}{\partial\mathbf{w}_j} \tag{17}$$

for a given input vector $\mathbf{v}$ and $\mathbf{w}_j \in W^-(\mathbf{v})$, i.e. all incorrect prototypes are gradually moved towards the correct best matching prototype $\mathbf{w}^+$ according to their dissimilarity rank with respect to $\mathbf{w}^+$. For decreasing neighborhood range, as it is usual in learning, only the incorrect prototype with smallest dissimilarity to $\mathbf{w}^+$ is attracted in the limit $\sigma_- \searrow 0$.

Summarizing we can state that σ_- adjusts the neighborhood cooperativness while the weighting coefficient γ controls the influence of border sensitive learning in this model.

3.3 Kernel GLVQ and Class Border Sensitive Learning as an Alternative to SVMs

Obviously, both proposed methods of class border sensitive learning can be combined with KGLVQ from sec. 2.2: The parametrized transfer function f_θ of the GLVQ cost function (1) takes the underlying dissimilarity measure involved in the classifier function μ from (2) into account. However, now explicit new dependence is installed such that kernel distances are immediately applicable also for this kind of class border sensitive learning.

For the second method, the dissimilarity measure is explicitly occurring in the penalty force $F_{neigh}(W, V)$ from (14). This penalty term compel an additional update term for prototype updates (17) containing, hence, the derivative of the applied dissimilarity measure. Yet, in KGLVQ the dissimilarity is the kernel distance kernel d_{κ_Φ} from (9). Obviously, it is differentiable if the kernel is assumed to be differentiable. Therefore, there is no restriction for border sensitive learning in case of universal differentiable kernel.

Summarizing, both methods for border sensitive learning can be combined with GLVQ. Thus, prototypes in KGLVQ are enabled to detect class borders in the kernelized data space $\mathcal{V}$ like SVMs in the function space $\mathcal{I}_{\kappa_\oplus}$. However, the prototypes are neither data vectors itself like support vectors in SVM nor exact class border vectors. They remain an average over the local data points surrounding the class borders. A further difference to SVMs is the model complexity, which is automatically predefined by the choice of the number of prototypes used for each class. As pointed out above, SVMs may occupy many data to serve as support vectors. In the worst case all data become support vectors. KGLVQ without any growing strategy would deliver reduced accuracy in case of underestimated number of needed prototypes. The restriction to prototypes being data vectors could be realized applying median variants of LVQ [12]. An extension of this idea to GLVQ is topic of current research.

4 Illustrative Simulations

In the following we give some illustrative examples for the above introduced concepts. All data sets are two-dimensional for better visualization.

The first example demonstrates the influence of the parameter θ of the logistic function f_θ in the cost function $E_{GLVQ}(W)$ from (1). We consider a two-class-problem, see Fig.4. Depending on the chosen value θ, the active set $\hat{\Xi}$ varies and

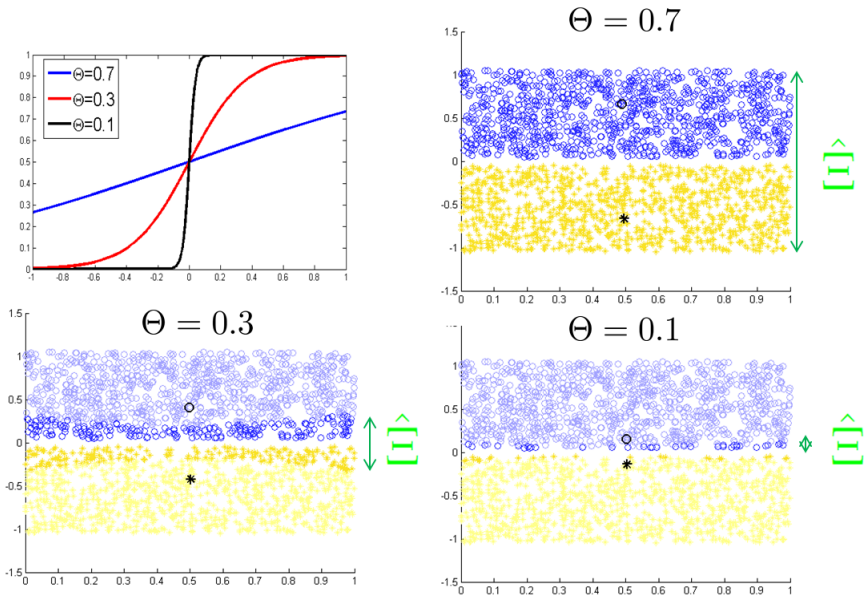


Figure 4: Visualization of influence of the parameter θ of the logistic function f_θ in the cost function $E_{GLVQ}(W)$. Depending on θ, the active set $\hat{\Xi}$ becomes thin or thick and the prototypes (black symbols) move accordingly.

the prototypes are adapted accordingly. We can observe, that for smaller values of θ the active set becomes smaller and, hence, the prototypes are localized closer to the border.

In the second example we consider the BS-GLVQ for a three-class problem. In the first simulation of this task only one prototype is used per class. If we have a non-vanishing penalty influence, i.e. $\gamma > 0$ in the cost function (13), the prototypes are moved to the class borders whereas for $\gamma = 0$, which is equivalent to standard GLVQ, the prototypes are positioned approximately in the class centers, see Fig. 5. If more than one prototype per class are used, the prototypes are placed close to the class borders as well as in the inner class regions, see Fig. 6. Hence, the slowly vanishing neighborhood cooperativeness according to (15) with decreasing range $\sigma_- \searrow 0$ during the adaptation process, distributes the prototypes close to the class borders as well as into the class centers.

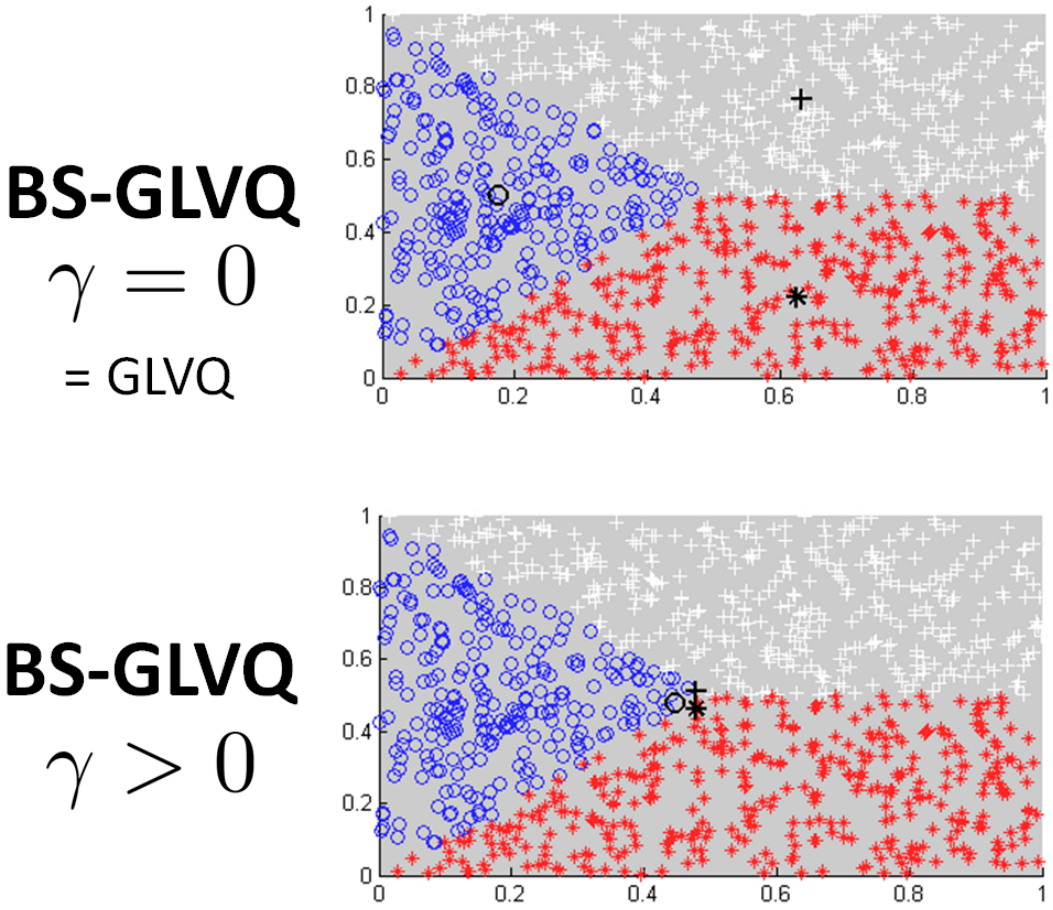


Figure 5: Visualization of influence of the penalty term $F_{neigh}(W, V)$ in the cost function $E_{BS-GLVQ}(W, \gamma)$ controlled by γ. Only one prototype per class is allowed.

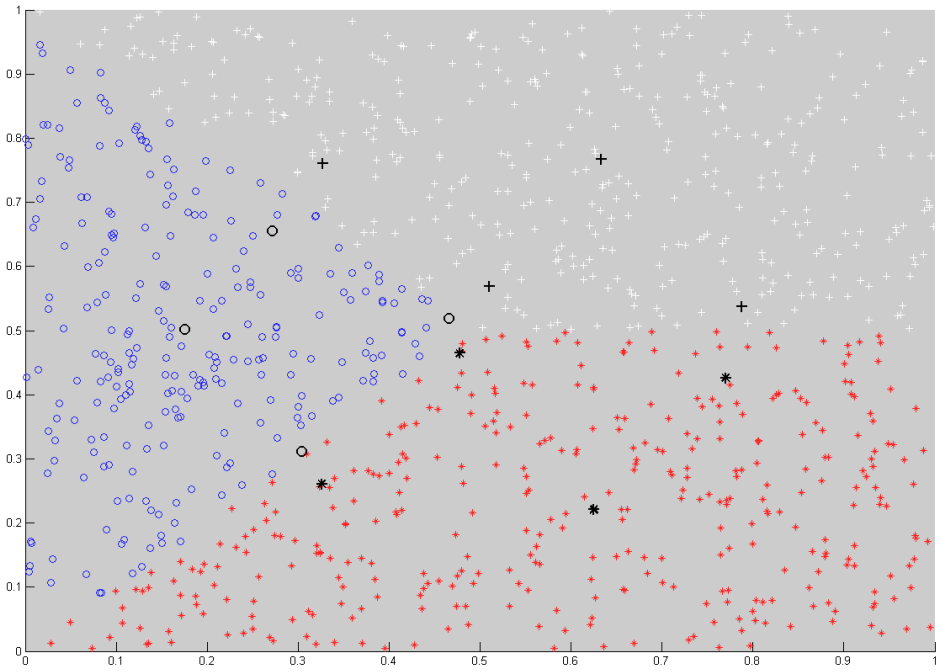


Figure 6: Visualization of border sensitive learning using the penalty term $F_{neigh}(W, V)$ with 4 prototypes per class (black symbols). Three prototypes per class detect the class borders whereas one prototype for each class is responsible for the inner class areas.

5 Conclusion and Outlook

In the present contribution we considered two possibilities for class border sensitive learning in GLVQ whereas original GLVQ adjusts the prototypes in the center of the classes. This is in contrast to SVMs, where the support vectors (prototypes) represent the class borders. The first possibility to achieve such a behavior in GLVQ uses the parametrized logistic function as transfer function in GLVQ, whereby the parameter controls the strength of the class border sensitivity. The second approach utilizes an additive penalty term in the cost function of GLVQ to obtain class border sensitivity. We have shown illustrative examples that both approaches deliver the expected results. An advantage of the introduced approaches compared to SVM is the explicit control of the model complexity, because the number of prototypes has to be chosen in advance whereas in SVMs the number of support vector may become quite large in case of difficult classification tasks. Hence, the provided extensions of GLVQ can be seen as an alternative to SVM, in particular, if border sensitivity is combined with KGLVQ, because it offers a good control over the model complexity while offering both border-sensitive prototypes and class-representing prototypes at the same time. Additionally, the range of neighborhood cooperativness in BS-(K)GLVQ allows a control of the amount of prototypes to be sensitive for class borders.

References

[1] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

[2] 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(1):159–173, 2012.

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

[4] T. Geweniger, M. Kästner, and T. Villmann. Border sensitive classification learning in fuzzy vector quantization. *Machine Learning Reports*, 6(MLR-04-2012):in press, 2012. ISSN:1865-3960, http://www.techfak.uni-bielefeld.de/~fschleif/mlr/mlr_04_2012.pdf.

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

[6] S. Haykin. *Neural Networks. A Comprehensive Foundation*. Macmillan, New York, 1994.

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

[8] 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.

[9] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society, London, A*, 209:415–446, 1909.

[10] E. Mwebaze, P. Schneider, F.-M. Schleif, J. Aduwo, J. Quinn, S. Haase, T. Villmann, and M. Biehl. Divergence based classification in learning vector quantization. *Neurocomputing*, 74(9):1429–1435, 2011.

[11] E. Pekalska and R. Duin. *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications*. World Scientific, 2006.

[12] I. Pitas, C. Kotropoulos, N. Nikolaidis, R. Yang, and M. Gabbouj. Order statistics learning vector quantizer. *IEEE Transactions on Image Processing*, 5(6):1048–1053, 1996.

[13] 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.

[14] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

[15] P. Schneider, M. Biehl, and B. Hammer. Hyperparameter learning in robust soft LVQ. In M. Verleysen, editor, *Proceedings of the European Symposium on Artificial Neural Networks ESANN*, pages 517–522. d-side publications, 2009.

[16] 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.

[17] P. Schneider, B. Hammer, and M. Biehl. Adaptive relevance matrices in learning vector quantization. *Neural Computation*, 21:3532–3561, 2009.

[18] P. Schneider, B. Hammer, and M. Biehl. Distance learning in discriminative vector quantization. *Neural Computation*, 21:2942–2969, 2009.

[19] S. Seo, M. Bode, and K. Obermayer. Soft nearest prototype classification. *IEEE Transaction on Neural Networks*, 14:390–398, 2003.

[20] S. Seo and K. Obermayer. Soft learning vector quantization. *Neural Computation*, 15:1589–1604, 2003.

[21] S. Seo and K. Obermayer. Dynamic hyperparameter scaling method for LVQ algorithms. In *Proc. of the International Joint Conference on Neural Networks (IJCNN'06)*, pages 3196 – 3203. IEEE Press, 2006.

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

[23] I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2001.

[24] M. Strickert. Enhancing M|G|RLVQ by quasi step discriminatory functions using 2^{nd} order training. *Machine Learning Reports*, 5(MLR-06-2011):5–15, 2011. ISSN:1865-3960, http://www.techfak.uni-bielefeld.de/~fschleif/mlr/mlr_06_2011.pdf.

[25] T. Villmann, T. Geweniger, M. Kästner, and M. Lange. Theory of fuzzy neural gas for unsupervised vector quantization. *Machine Learning Reports*, 5(MLR-06-2011):27–46, 2011. ISSN:1865-3960, http://www.techfak.uni-bielefeld.de/~fschleif/mlr/mlr_06_2011.pdf.

[26] T. Villmann, T. Geweniger, M. Kästner, and M. Lange. Fuzzy neural gas for unsupervised vector quantization. In L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. Zadeh, and J. Zurada, editors, *Artificial Intelligence and Soft Computing - Proc. the International Conference ICAISC, Zakopane*, volume 1 of *LNAI 7267*, pages 350–358, Berlin Heidelberg, 2012. Springer.

[27] T. Villmann and S. Haase. Divergence based vector quantization. *Neural Computation*, 23(5):1343–1392, 2011.

[28] T. Villmann and S. Haase. A note on gradient based learning in vector quantization using differentiable kernels for Hilbert and Banach spaces. *Machine Learning Reports*, 6(MLR-02-2012):1–29, 2012. ISSN:1865-3960, http://www.techfak.uni-bielefeld.de/~fschleif/mlr/mlr_02_2012.pdf.

[29] A. Witoelar, A. Gosh, J. de Vries, B. Hammer, and M. Biehl. Window-based example selection in learning vector quantization. *Neural Computation*, 22(11):2924–2961, 2010.

[30] C. Yin, S. Mu, and S. Tian. Using cooperative clustering to solve multi-class problems. In Y. Wang and T. Li, editors, *Foundation of Intelligent Systems - Proc. of the Sixth International Conference on Intelligent Systems and Knowledge Engineering (ISKE 2011), Shanghei, China*, volume 122 of *Advances in Intelligent and Soft Computing*, pages 327–334. Springer, 2012.

Accelerated Vector Quantization by Pulsing Neural Gas

Lydia Fischer, Mandy Lange, Marika Kästner, Thomas Villmann
University of Applied Sciences Mittweida,
Technikumplatz 17, 09648 Mittweida, Germany

Abstract

In this article we introduce a new variant of neural vector quantization based on neural gas. It is a combination of standard neural gas vector quantizer with a simulated annealing approach while keeping the rank-based approach. Especially, for the batch version of neural gas, which usually converges rapidly but have a strong tendency to stuck in local optima, this might be a strategy overcome these difficulties.

1 Introduction

Clustering and vector quantization is still a challenging task requiring powerful algorithms, in particular, if large data sets have to be processed. Neural vector quantizer like self-organizing maps (SOM,[8]) or neural gas (NG,[10]) are powerful prototype-based methods for fast and accurate data analysis. The Heskes-variant of SOM [5] as well as NG realize a stochastic gradient descent on a cost function related to the squared description error. However, stochastic gradient approaches usually require long-time training processes according to the underlying theory [3, 9, 11]. Faster learning may cause suboptimal solution because the algorithms stuck in local minima. This property is frequently observed for the batch variants. A strategy to avoid this behavior is simulated annealing [7].

In the present contribution we combine the NG vector quantizer with ideas adopted from simulated annealing, i.e. we introduce a reverse learning: It is carried out with some probability during learning to achieve a temporarily deterioration of the prototype configuration, which increases the probability to abandon a local

optimum. For this purpose, we briefly revisit the NG. Thereafter, we introduce the new reverse learning obtaining the so-called *Pulsing NG*.

2 Neural Gas

For the neural gas, we consider a data set $\mathbf{V}$ with data points $\mathbf{v}$ and we would like to have a set of prototypes $\mathbf{W} = \{\mathbf{w}_j | j = 1, \ldots, N\}$ which represent the data set $\mathbf{V} = \{\mathbf{v}_k | j = 1, \ldots, M\}$. The NG minimizes the cost function

$$E_{NG} = \frac{1}{2 \cdot C(\lambda)} \int \sum_{j=1}^{N} P(\mathbf{v}) \cdot h_\lambda(rg_j(\mathbf{v}, \mathbf{W})) \cdot (\mathbf{v} - \mathbf{w}_j)^2 d\mathbf{v} \tag{1}$$

by means of stochastic gradient descend learning [10]. Here,

$$d(\mathbf{v}, \mathbf{w}_j) = (\mathbf{v} - \mathbf{w}_j)^2$$

is the squared Euclidean distance and $P(\mathbf{v})$ is the data point density. The function $rg_j(\mathbf{v}, \mathbf{W}) \in \{0, \ldots, N-1\}$ quotes the position of each prototype $\mathbf{w}_j$ according to the data point $\mathbf{v}$. It can be calculated in the following way

$$rg_j(\mathbf{v}, \mathbf{W}) = \sum_{i=1}^{N} H(d(\mathbf{v}, \mathbf{w}_j) - d(\mathbf{v}, \mathbf{w}_i)) \tag{2}$$

and $H(x)$ is the Heaviside-function. According to $rg_j(\mathbf{v}, \mathbf{W})$ we introduce a neighborhood function $h_\lambda(rg_j(\mathbf{v}, \mathbf{W}))$. It is defined as

$$h_\lambda(rg_j(\mathbf{v}, \mathbf{W})) = e^{\left(-\frac{rg_j(\mathbf{v}, \mathbf{W})}{\lambda}\right)}. \tag{3}$$

The prototypes $\mathbf{w}_j$ are updated correspondingly to the stochastic gradient decent on E_{NG} (1) as

$$\mathbf{w}_j = \mathbf{w}_j - \epsilon \frac{\partial E_{NG}}{\partial \mathbf{w}_j} \tag{4}$$

with

$$\frac{\partial E_{NG}}{\partial \mathbf{w}_j} \sim -h_\lambda(rg_j(\mathbf{v}, \mathbf{W})) \cdot (\mathbf{v} - \mathbf{w}_j). \tag{5}$$

In each update step, a randomly selected data point $\mathbf{v}$ is presented according to the data distribution $P(\mathbf{v})$. Then the prototypes are updated by means of (4). This update rule (4) is a '*soft-max*' adaption because not only the closest prototype to

the data point is updated assort all prototypes get an update according to their position $rg_j(\mathbf{v}, \mathbf{W})$. The number of prototypes taken effecctively into account during the update can be controlled by the neighborhood range λ.

In [1] an advanced version of the NG, the Batch NG (BNG) was presented, which converges faster. In each iteration step all data points are considered and therefore all prototypes are updated. The update rule for the prototypes is the following:

$$\mathbf{w}_j = \frac{\sum_{k=1}^{M} h_\lambda(rg_j(\mathbf{v}, \mathbf{W})) \cdot \mathbf{v}_k}{\sum_{k=1}^{M} h_\lambda(rg_j(\mathbf{v}, \mathbf{W}))}. \tag{6}$$

However, this BNG shows a strong tendency to stuck in local minima. This is demonstrated for the well-known two-dimensional (multimodal) checkerboard data set from [4], visualized in Fig. 1.

3 A Simulated Annealing Modification of Neural Gas Algorithm

In the following we discuss a strategy to improve the convergence behavior of BNG adopting ideas from Simulated Annealing (SA) [7]. First we briefly review SA. After this we consider its integration into NG/BNG.

3.1 Simulated Annealing

An effective strategy of optimization is the heuristic SA [7]. The idea of SA comes from thermodynamics such that a deterioration during the optimization process is accepted with a certain probability. This strategy allows to leave local optima in order to find the global one. In particular, let us assume a cost function $f(\mathbf{x}) \to \mathbb{R}$ and a set $\mathbf{X}$ of feasible solutions. Without loss of generality the cost function has to be minimized. We start with a feasible solution $\mathbf{x} \in \mathbf{X}$ and create a neighborhood $N(\mathbf{x}) \subseteq \mathbf{X}$ related to this solution. Afterwards another solution $\mathbf{x}^{new} \in N(\mathbf{x})$ is picked. It will be immediately accepted, if $f(\mathbf{x}^{new}) < f(\mathbf{x})$ is valid. Otherwise, it will be accepted only with some probability

$$p(\Delta, T) = e^{\left(-\frac{\Delta}{T}\right)}. \tag{7}$$

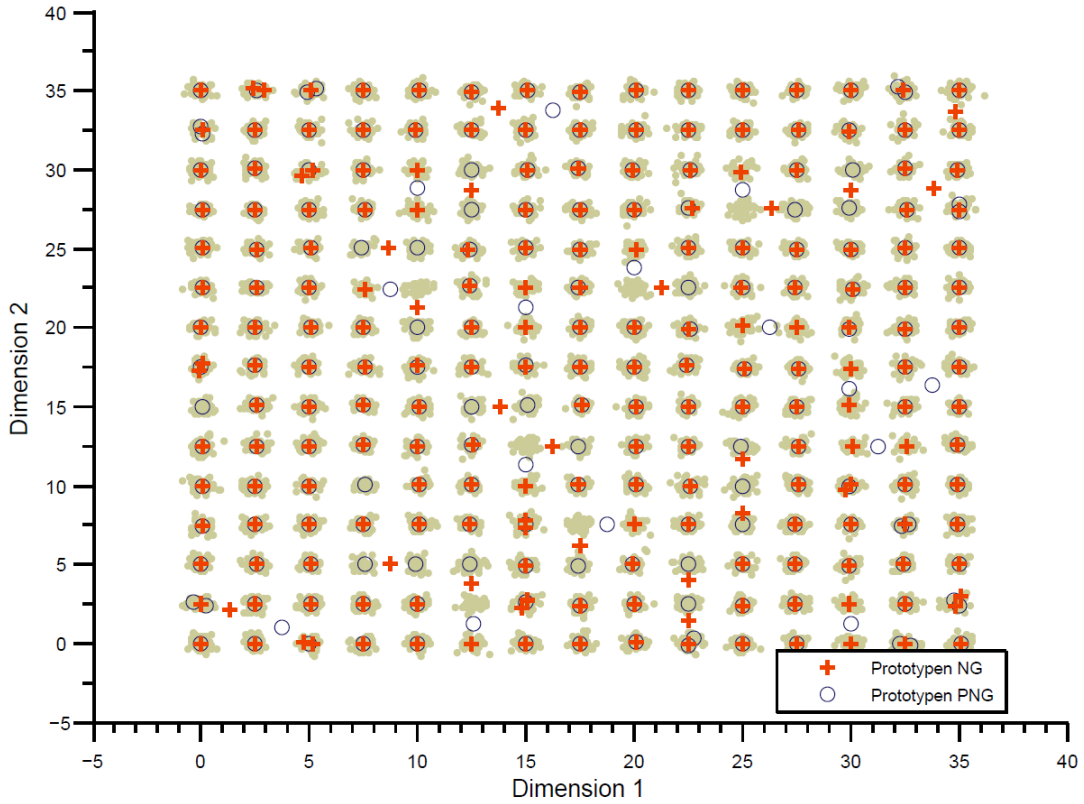


Figure 1: Clustering of a (multimodal) checker board data set by NG (batch) and PNG (batch). The NG-variants use as many prototypes as clusters in the checkerboard. The PNG achieves better results than NG: the latter one shows more inaccurate prototypes.

with $\Delta = f(\mathbf{x}^{new}) - f(\mathbf{x})$. In fact, $p(\Delta, T)$ is chosen as a Boltzmann-Gibbs probability with the 'temperature' T, which is slowly decreased during time. Thus, the algorithm temporarily approves worse solutions with decreasing probability.

3.2 Integration of Simulated Annealing into Neural Gas – Pulsing Neural Gas

The Pulsing Neural Gas (PNG) is a combination of NG and SA. In the following we denote a time step in which an iterative algorithm generates an definitely worse solution as a *negative learning* or *negative learning step*. In terms of the NG cost function E_{NG} (1) a negative learning would yield an increasing cost function value. In case of stochastic gradient learning a descent is not guaranteed in each learning step. Hence, the character, positive or negative, has to be determined separately based on the evaluation of E_{NG}. Of course, in average, i.e. with high probability, we can assume positive learning for usual NG.

Negative learning would correspond to an acceptance of a deterioration in SA. A possibility for averaged negative learning would be to apply a gradient ascent step with some probability. This probability should follow the Boltzmann-Gibbs distribution as in SA. However, investigations have shown that this strategy cause unstable learning [2]. Therefore, we suggest another way instead of a gradient ascent: According to the Gibbs-probability a *reverse ranking*

$$rg_j^-(\mathbf{v}, \mathbf{W}) = (N-1) - \sum_{i=1}^{N} H(d(\mathbf{v}, \mathbf{w}_j) - d(\mathbf{v}, \mathbf{w}_i)). \tag{8}$$

of the prototypes for a given data point $\mathbf{v}$ is applied. It reverses the usual (positive) ranking from (2) in such a way that here the prototype with the largest distance becomes the best zero-rank, see Fig.2.

Then, a *reverse learning step* in online NG is executed as

$$\mathbf{w}_j = \mathbf{w}_j + \epsilon \frac{\partial E_{NG}^-}{\partial \mathbf{w}_j}. \tag{9}$$

whereby E_{NG}^- is the cost function of NG but with the new negative rank function (8). Analogously, a reverse learning in BNG is accomplished by

$$\mathbf{w}_j = \frac{\sum\limits_{\mathbf{v} \in \mathbf{A}} h_\lambda(rg_j^-(\mathbf{v}, \mathbf{W})) \cdot \mathbf{v}}{\sum\limits_{\mathbf{v} \in \mathbf{A}} h_\lambda(rg_j^-(\mathbf{v}, \mathbf{W}))}. \tag{10}$$

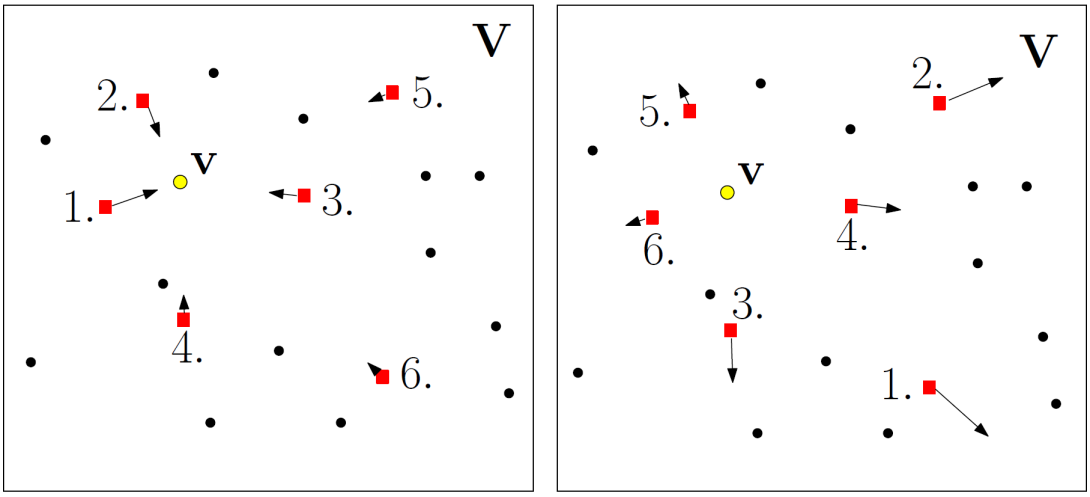


Figure 2: Comparison of the positive and negative ranking in NG for a given data vector $\mathbf{v}$ and resulting adaptation shifts: left - usual (positive) ranking in NG according to the rank function $rg_j(\mathbf{v}, \mathbf{W})$ from (2) with usual NG-learning; right - negative ranking according to $rg_j^-(\mathbf{v}, \mathbf{W})$ from (8) with reverse learning. Prototypes are depicted as red squares and data points as black dots.

whereas $\mathbf{A} \subset \mathbf{V}$ is an non-empty subset.

In average, reverse learning leads to negative learning and, hence, realizes the SA-strategy. The resulting *pulsing NG* (PNG) algorithm is summarized in Alg. 1.

A careful reverse learning (10) has a strong influence on the realization of a severe degradation. To avoid too heavy distortions one can use a convex linear combination of usual NG-learning and reverse learning [2].

Application of the PNG to the above checkerboard problem delivers better results, see Fig. 1. The development in time of the NG cost function E_{NG} is depicted in Fig. 3 for BNG and (batch) PNG.

We clearly observe the improved convergence behavior.

4 Conclusion

In this contribution we introduce the pulsed neural gas (PNG) as an alternative to standard NG. It incorporates ideas of simulated annealing into NG to avoid local minima. This is is of particular interest, if batch variants are applied, which frequently cause only local optimality. It should be noted at this point that PNG

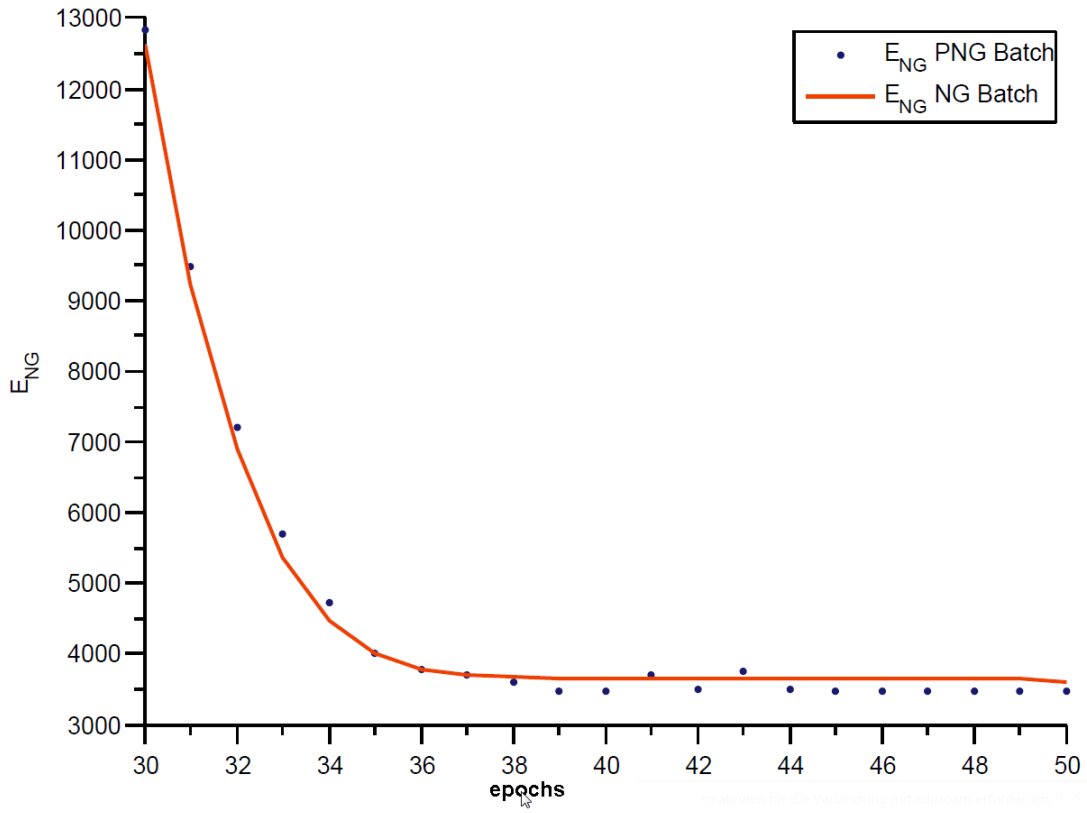


Figure 3: Clustering of a checker board data set by NG (batch) and PNG (batch). The PNG achieves better results than NG, which shows more inaccurate prototypes.

Algorithm 1 PNG

Input: data set $\mathbf{V}$, number of prototypes

Output: prototypes $\mathbf{w}_j$

```
initialization
for T = 0 to training steps do
  determine p(T), uniformly distributed random number z
  if p(T) < z then
    original NG step
  else
    reverse learning step
  end if
end for
```

is not restricted to the Euclidean distance. Obviously, other differentiable dissimilarity measures like divergences or kernel distance may be applied replacing the respective derivatives accordingly [6, 12, 13, 14].

References

[1] M. Cottrell, B. Hammer, A. Hasenfuß, and T. Villmann. Batch and median neural gas. *Neural Networks*, 19:762–771, 2006.

[2] L. Fischer. Modifikation unüberwachter Vektorquantisierer für funktionale Daten und Einbindung einer neuen Optimierungsstrategie. Master's thesis, University of Applied Sciences Mittweida, Mittweida, Saxony, Germany, 2012.

[3] S. Graf and H. Lushgy. *Foundations of quantization for random vectors*. LNM-1730. Springer, Berlin, 2000.

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

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

[6] M. Kästner, B. Hammer, M. Biehl, and T. Villmann. Functional relevance learning in generalized learning vector quantization. *Neurocomputing*, 90(9):85–95, 2012.

[7] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

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

[9] H. Kushner and D. Clark. *Stochastic Appproximation Methods for Constrained and Unconstrained Systems*. Springer-Verlag, New York, 1978.

[10] 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.

[11] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Stat.*, 22:400–407, 1951.

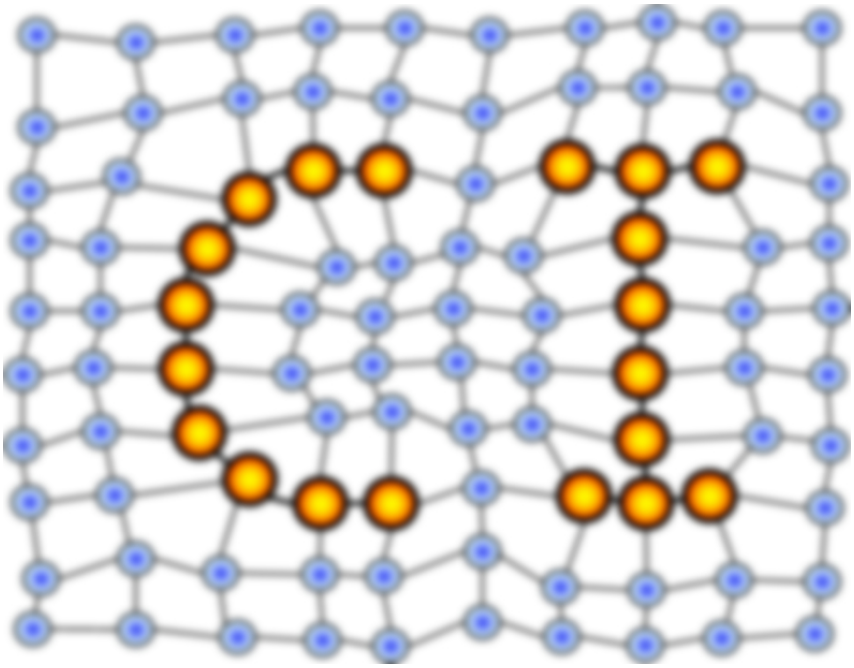
[12] T. Villmann. Sobolev metrics for learning of functional data - mathematical and theoretical aspects. *Machine Learning Reports*, 1(MLR-03-2007):1–15, 2007. ISSN:1865-3960, http://www.uni-leipzig.de/~compint/mlr/mlr_01_2007.pdf.

[13] T. Villmann and S. Haase. Divergence based vector quantization. *Neural Computation*, 23(5):1343–1392, 2011.

[14] T. Villmann, S. Haase, and M. Kästner. Gradient based learning in vector quantization using differentiable kernels. In P. Estevez, J. Principe, and P. Zegers, editors, *Advances in Self-Organizing Maps: 9th International Workshop WSOM 2012 Santiage de Chile*, volume 198 of *Advances in Intelligent Systems and Computing*, pages 193–204, Berlin, 2012. Springer.

MACHINE LEARNING REPORTS

Report 06/2012



Impressum

Machine Learning Reports ISSN: 1865-3960

▽ Publisher/Editors

Prof. Dr. rer. nat. Thomas Villmann
University of Applied Sciences Mittweida
Technikumplatz 17, 09648 Mittweida, Germany
• http://www.mni.hs-mittweida.de/

Dr. rer. nat. Frank-Michael Schleif
University of Bielefeld
Universitätsstrasse 21-23, 33615 Bielefeld, Germany
• http://www.cit-ec.de/tcs/about

▽ Acknowledgments

We would like to thank the reviewers for their time and patience.