DE GRUYTER OPEN

JAISCR, 2017, Vol. 7, No. 1, pp. 65– 81
10.1515/jaiscr-2017-0005

CAN LEARNING VECTOR QUANTIZATION BE AN ALTERNATIVE TO SVM AND DEEP LEARNING? - RECENT TRENDS AND ADVANCED VARIANTS OF LEARNING VECTOR QUANTIZATION FOR CLASSIFICATION LEARNING

Thomas Villmann[1], Andrea Bohnsack[1,2], Marika Kaden[1]

[1] *Computational Intelligence Group,*
University of Applied Sciences Mittweida, Germany
email: thomas.villmann@hs-mittweida.de

[2]*Staatliche Berufliche Oberschule Kaufbeuren, Germany*

Abstract

Learning vector quantization (LVQ) is one of the most powerful approaches for prototype based classification of vector data, intuitively introduced by Kohonen. The prototype adaptation scheme relies on its attraction and repulsion during the learning providing an easy geometric interpretability of the learning as well as of the classification decision scheme. Although deep learning architectures and support vector classifiers frequently achieve comparable or even better results, LVQ models are smart alternatives with low complexity and computational costs making them attractive for many industrial applications like intelligent sensor systems or advanced driver assistance systems.

Nowadays, the mathematical theory developed for LVQ delivers sufficient justification of the algorithm making it an appealing alternative to other approaches like support vector machines and deep learning techniques.

This review article reports current developments and extensions of LVQ starting from the generalized LVQ (GLVQ), which is known as the most powerful cost function based realization of the original LVQ. The cost function minimized in GLVQ is an soft-approximation of the standard classification error allowing gradient descent learning techniques. The GLVQ variants considered in this contribution, cover many aspects like border-sensitive learning, application of non-Euclidean metrics like kernel distances or divergences, relevance learning as well as optimization of advanced statistical classification quality measures beyond the accuracy including sensitivity and specificity or area under the ROC-curve.

According to these topics, the paper highlights the basic motivation for these variants and extensions together with the mathematical prerequisites and treatments for integration into the standard GLVQ scheme and compares them to other machine learning approaches. For detailed description and mathematical theory behind all, the reader is referred to the respective original articles.

Thus, the intention of the paper is to provide a comprehensive overview of the state-of-the-art serving as a starting point to search for an appropriate LVQ variant in case of a given specific classification problem as well as a reference to recently developed variants and improvements of the basic GLVQ scheme.

Keywords: classification learning, vector quantization, prototype based learning

1 Introduction

Classification learning from data samples is one of the most important tasks in computational intelligence. It belongs to the supervised learning approaches, which is the most common form in machine learning [1]. Several strategies are established for this task ranging from multi-layer perceptrons (MLP,[2, 3]), statistical learning techniques to deep learning (DL) architectures, which currently gain large attraction [4, 5]. Yet, all these approaches are mainly influenced by the Bayes theory of decisions and classification [6, 7].

Among general supervised learning problems, the classification learning of vector data plays a central role in data processing. A huge amount of data are given as vectors or can be coded in vectorial form after preprocessing. For those kinds of data, classification schemes designed especially for vector data are frequently more appropriate than general schemes like decision trees taking not into account the vector structure of the data [8]. Moreover, keeping in mind the rapidly growing number of available data, classification learning is demanded to handle big data leading to the related task of data compression. However, classification learning and data compression of vector data are the two different sides of one coin, which are closely related to supervised and unsupervised vector quantization [9]. Beside other paradigms, prototype based approaches are known to be robust machine learning methods with high performance, in general, while being often easy to interpret [10]. In this sense, unsupervised vector quantization consists of the generation of a set of representative prototype vectors for a given data set, whereby the distribution of the prototypes should approximate the data density as good as possible [11]. The approaches differ in the interpretation of the approximation criteria. Here geometric approaches like in c-means [12] or general information theoretic approaches are favored [13, 14]. Biologically inspired vector quantization learning approaches like the self-organizing map (SOM) or the neural gas (NG) vector quantizer are effective tools for the vector quantization task [15, 16]. In case of supervised vector quantization for classification learning, the prototypes are used to determine the classification decision. Two basic principles can be distinguished: class representative prototypes are demanded to represent the class distributions whereas border-sensitive schemes favor prototypes indicating the class borders. One of the most prominent examples of the latter strategy is the support vector machine approach (SVM,[17]), where the prototypes are denoted as support vectors. In contrast, original learning vector quantization (LVQ) as introduced by T. Kohonen was designed to approximate the class distributions by representative prototypes as known from unsupervised vector quantization [18, 19, 20]. The intuitive interpretation and the robust behavior of LVQ have gained a large attraction of this neural based learning approach [21, 22]. Yet, the basic LVQ models are restricted in the range of applications, because many more complex classification tasks can not be handled in adequate manner by the original scheme. During the last years these drawbacks were heavily attacked by the community. Extensions and modifications of the original LVQ were established to overcome many difficulties while keeping the basic principles and ideas. Moreover, mathematical justification of the algorithm variants and their properties lead a theoretical basis providing a framework comparable to statistical learning theory for SVM.

Another important aspect of LVQ models, making them interesting for industrial applications or intelligent sensor systems like advanced driver assistance systems, is their pre-determined complexity due to the fixed number of prototypes. This property can be a disadvantage, which can be partially solved by growing architectures known from unsupervised learning [23, 24]. However, this may cause instabilities known as the stability-plasticity-dilemma [25]. Otherwise, the pre-defined number maybe an advantage for restricted environments. If the performance is lower than for complex networks like deep learning models with a large number of pre-trained sub-layers or SVM models with a huge number of support vectors as we frequently find for high-performance systems for difficult recognition tasks, but the performance is still sufficient for a smart LVQ model, the latter one is the preferred alternative for systems with restricted resources.

This review article outlines these developments in theory of LVQ and relates them to other classification learning strategies like SVM and DL. Yet, rather being a complete book of LVQ theory it is more a good starting point for more detailed study in the field. Hence, it can be more seen as a guid-

ance how to adapt the original LVQ to new problems and applications.

2 The origins of LVQ and the basic standard scheme

For LVQ we suppose training data $V \subseteq \mathbb{R}^n$ with each $\mathbf{v} \in V$ has a class label $c(\mathbf{v}) \in \mathcal{C} = \{1,\ldots,C\}$ indicating to which class $\mathbf{v}$ belongs. Further, we assume M prototypes $W = \{\mathbf{w}_k \in \mathbb{R}^n, k = 1\ldots M\}$ with labels $c(\mathbf{w}_k) \in \mathcal{C}$ such that at least one prototype is assigned to each class.

2.1 The origins of LVQ - Bayesian motivation of the attraction-repusion-scheme

We start with the motivation of LVQ based on the Bayes theory of decisions and general vector quantization, as it is proposed by Kohonen [15]. His main observation is that for the realization of such a classifier the estimation of the class densities could be realized by means of unsupervised vector quantization using class related data densities. To explain this idea we follow exactly the explanations in [15] but with more mathematical precision:

Suppose a class probability model with classes $\mathcal{C} = \{1,\ldots,C\}$ defined in $\mathbb{R}^n$, where the priori probability of class $c \in C$ is denoted as P_c. The probability that a vector $\mathbf{x} \in \mathbb{R}^n$ is generated by class c is the conditional model probability $P(\mathbf{x}|c)$ such that $P(\mathbf{x}) = \sum_c P(\mathbf{x}|c)$ is the overall model density function.

We further assume that given training data $\mathbf{v} \in V$ are generated by this model. Hence, the conditional data probability $P(\mathbf{v}|c)$ takes $P(\mathbf{v}|c) = 1$ iff $c(\mathbf{v}) = c$ and zero elsewhere.

Following the Bayes theory we consider the model discriminant function

$$\delta_c(\mathbf{x}) = P(\mathbf{x}|c) \cdot P_c, \tag{1}$$

with

$$\delta_{c^*}(\mathbf{x}) = \max_{c \in C} \{\delta_c(\mathbf{x})\}, \tag{2}$$

determines the optimum decision. The Bayesian model class region B_c of the class c regarding the probability model consists of those vectors $\mathbf{x} \in \mathbb{R}^n$ for which $c^* = c$ is valid, i.e. for which the class determining function

$$b_c(\mathbf{x}) = \begin{cases} \frac{\delta_c(\mathbf{x}) - \delta_{h^*}(\mathbf{x})}{\beta} & \text{if } \mathbf{x} \in B_c \\ 0 & \text{if } \mathbf{x} \notin B_c \end{cases}, \tag{3}$$

of the probabilistic model is greater than zero for an arbitrary normalization constant $\beta > 0$. Here

$$h^* = argmax_{h \in C \setminus \{c\}} \{\delta_h(x)\},$$

is the most proximate discriminant function of a class h different from c (incorrect class with respect to the Bayesian model class region B_c). Thus the sum

$$b(\mathbf{x}) = \sum_{c \in C} b_c(\mathbf{x}), \tag{4}$$

with β from (3) chosen as $\beta = \int b(\mathbf{x}) d\mathbf{x}$ becomes a formal class model density function, which can be taken as a Bayes decision based class probability density related to the given class probabilistic model, which vanishes at the Bayesian class borders.

In unsupervised prototype based vector quantization the N prototypes $W = \{\mathbf{w}_k \in \mathbb{R}^n, k = 1\ldots M\}$ without label information should represent V as good as possible [11]. The winner-takes-all rule (WTA)

$$s(v) = argmin_{k=1,\ldots,M} d(v, w_k), \tag{5}$$

realizes a nearest prototype principle, such that $\mathbf{w}_{s(\mathbf{v})}$ is the overall winner prototype with $d^s(\mathbf{v}) = d\left(\mathbf{v}, \mathbf{w}_{s(\mathbf{v})}\right)$. Here d is a general dissimilarity measure, usually chosen as the squared Euclidean distance d_2 being the special case $p = 2$ for the p-th power

$$d_p(\mathbf{v}, \mathbf{w}) = \sum_{i=1}^{n} |v_i - w_i|^p, \tag{6}$$

of the Minkowski l_p-distance. The expected quantization error

$$E = \int_V d_2\left(\mathbf{v}, \mathbf{w}_{s(\mathbf{v})}\right) P(\mathbf{v}) d\mathbf{v}, \tag{7}$$

depends on the overall model density $P(\mathbf{x})$ evaluated for the data vector $\mathbf{v}$. This error can be optimized by stochastic gradient descent learning (SGDL) yielding

$$\nabla_{\mathbf{w}_k} E = -2 \int \delta_{k,s(\mathbf{v})} (\mathbf{v} - \mathbf{w}_k) P(\mathbf{v}) d\mathbf{v}, \tag{8}$$

as averaged update rule [26, 27, 28]. The set

$$R_k = \{\mathbf{v} \in V | s(\mathbf{v}) = k\}, \tag{9}$$

is denoted as the receptive field of the prototype $\mathbf{w}_k$ or the (masked) Voronoi cell [29].

Now, the idea of Kohonen in context of the class dependend model is the formal replacement of the overall data density $P(\mathbf{v})$ in (7) by the class model density function b from (4) but here calculated for the data vectors $\mathbf{v}$ such that a class dependent vector quantizer is obtained with

$$E_{class-VQ} = \int_V d_2\left(\mathbf{v}, \mathbf{w}_{s(\mathbf{v})}\right) b(\mathbf{v})\, d\mathbf{v}, \qquad (10)$$

as the quantization error. Particularly, $b(\mathbf{v})$ depends on the values $P(\mathbf{v}|c)$ instead of the conditional model probabilities $P(\mathbf{x}|c)$ via the class dependent quantities $b_c(\mathbf{v})$. Further, let $s^* = s^*(\mathbf{v})$ be the index such that $\mathbf{v} \in B_{s^*(\mathbf{v})}$ is valid, i.e. $\mathbf{v}$ belongs to the Bayesian model class region $B_{s^*(\mathbf{v})}$. In complete analogy to the averaged update rule (8) we formally derive

$$\nabla_{\mathbf{w}_k} E_{class-VQ} = -2 \int \delta_{k,s(\mathbf{v})} (\mathbf{v} - \mathbf{w}_k)\, b(\mathbf{v})\, d\mathbf{v}$$
$$= -2 \int \delta_{k,s(\mathbf{v})} (\mathbf{v} - \mathbf{w}_k) \left(\frac{\delta_{s^*(\mathbf{v})} - \delta_{h^*(\mathbf{v})}}{\beta} \right) d\mathbf{v},$$

as a stochastic gradient. If $s^* = c(\mathbf{v})$, we obtain $\delta_{s^*}(\mathbf{v}) P(\mathbf{v}|s^*) \cdot P_{s^*} = P_{s^*}$ and $\delta_{h^*}(\mathbf{v}) = 0$, and hence $\mathbf{w}_{s(\mathbf{v})}$ is shifted towards the center of $B_{s^*(\mathbf{v})}$. Otherwise, if $s^* \neq c(\mathbf{v})$ it follows that $\delta_{h^*}(\mathbf{v}) = P_{c(\mathbf{v})}$ and $\delta_{s^*}(\mathbf{v}) P(\mathbf{v}|s^*) \cdot P_{s^*} = 0$ are valid leading to a repulsion punishment. In consequence, the (so far) unlabeled prototypes are asymptotically attracted to be responsible for the Bayesian model class regions.

These observations motivate the most basic learning rule known as LVQ1: As in the beginning of this chapter, we assume the prototypes equipped with class labels and the nearest prototype learning according to WTA (5). Both ingredients are used to estimate the Bayesian regions B_c based on the receptive fields R_k. More specifically, the goal is to obtain a good correspondence

$$B_c \approx \cup_k \{R_k | k = s(\mathbf{v}) \wedge c(\mathbf{w}_k) = c\},$$

obtained by prototype adaptation according to

$$\triangle \mathbf{w}_k = \alpha S(\mathbf{v}) (\mathbf{v} - \mathbf{w}_k),$$

with a learning rate $0 < \alpha \ll 1$ and shift control

$$S(\mathbf{v}) = \begin{cases} 1 & \text{if } s(\mathbf{v}) = k \wedge c(\mathbf{w}_k) = c(\mathbf{v}) \\ -1 & \text{if } s(\mathbf{v}) = k \wedge c(\mathbf{w}_k) \neq c(\mathbf{v}) \\ 0 & \text{else} \end{cases},$$

realizing the idea of attraction and repulsion punishment. Early improvements regarding an adaptive learning rate, update also of the second winner and stabilizing rules lead to variants OLVQ, LVQ2 and LVQ3. However, all these LVQ variants remain a heuristic. Moreover, although never explicitly stated so far, the quantity $E_{class-VQ}$ in (10) is not a valid cost function for vector quantization. This is due to the fact that for certain realizations $\mathbf{v}$ the evaluation of $b(\mathbf{v})$ may yield negative results and, hence, is not longer a density.

2.2 Mathematical justification of LVQ - the generalized LVQ as the basic standard scheme

Although original LVQ is only a heuristic method for classifier design, it is remarkably successful and delivers frequently very good results. As mentioned above, the resulting prototype adaptation rules allows a geometric interpretation in terms of attraction and repulsion, which is intuitive and simple. This easy but robust scheme is one of the key ingredients for the big popularity of LVQ. However, as we have seen before, a rigorous mathematical justification in terms of a *well-defined* cost function to be minimized is not available for the original LVQ.

To overcome this situation Sato&Yamada presented a generalized LVQ (GLVQ,[30]) minimizing a cost function based on an *approximation of the classification error* but keeping the principle of attraction and repulsion punishment. More precisely, they introduced a classifier function

$$\mu_d^W(\mathbf{v}) = \frac{d^+(\mathbf{v}) - d^-(\mathbf{v})}{d^+(\mathbf{v}) + d^-(\mathbf{v})}, \qquad (11)$$

with the quantities $d^+(\mathbf{v}) = d(\mathbf{v}, \mathbf{w}^+)$ and $d^-(\mathbf{v}) = d(\mathbf{v}, \mathbf{w}^-)$ for a general but differentiable dissimilarity measure d, which is usually chosen to be the squared Euclidean distance d_E as in the original version [30]. Here $\mathbf{w}^+ = \mathbf{w}^+(\mathbf{v})$ is the best matching prototype according to the WTA (5) but restricted to prototypes with correct class label $c(\mathbf{w}_k) = c(\mathbf{v})$, whereas $\mathbf{w}^- = \mathbf{w}^-(\mathbf{v})$ is the best matching prototype with incorrect class label. At this point it should be explicitly mentioned that $\mu_d^W(\mathbf{v})$ plays the role of a discriminant function in GLVQ comparable to (1) for Bayes classifier.

Obviously, the classifier function $\mu_d^W(\mathbf{v})$ yields

negative values in case of a correct classification. Hence, the overall exact classification error is given as

$$E_{LVQ}(V,W,d) = \frac{1}{\#V}\sum_{\mathbf{v}\in V} H\left(\mu_d^W(\mathbf{v})\right), \qquad (12)$$

with

$$H(x) = \begin{cases} 1 & \text{iff } x < 0 \\ 0 & \text{else} \end{cases}, \qquad (13)$$

being the Heaviside step function and $\#V$ denotes the cardinality of V. To ensure differentiability, Sato&Yamada replaced the Heaviside function by a monotonously increasing function f, frequently chosen either as the identity function $f_{\mathrm{id}}(x) = x$ or as a sigmoid function like

$$f_\theta(x) = \frac{1}{1+\exp\left(-\frac{x}{\theta}\right)}, \qquad (14)$$

where the parameter $\theta > 0$ controls the steepness of the slope, i.e. for $\theta \searrow 0$ the sigmoid function $f_\theta(x)$ becomes the Heaviside $H(x)$. Thus the cost function to be minimized for GLVQ rewrites as

$$E_{GLVQ}(V,W,f,d) = \frac{1}{\#V}\sum_{\mathbf{v}\in V} E(\mathbf{v},W,f,d), \qquad (15)$$

constituting an approximation of (12) with the local errors

$$E(\mathbf{v},W,f,d) = f\left(\mu_d^W(\mathbf{v})\right). \qquad (16)$$

SGDL for $E_{GLVQ}(V,W,d)$ is realized updating both, $\mathbf{w}^+$ and $\mathbf{w}^-$, at the same time for a randomly chosen input vector $\mathbf{v}$ according

$$\Delta\mathbf{w}^\pm = \alpha\xi_\mu^\pm(\mathbf{v},f,d)\frac{\partial d(\mathbf{v},\mathbf{w}^\pm)}{\partial\mathbf{w}^\pm}, \qquad (17)$$

with

$$\xi_\mu^\pm(\mathbf{v},f,d) = \frac{\pm 2d^\mp(\mathbf{v})\cdot f'(\mu(\mathbf{v}))}{\left(d^+(\mathbf{v})+d^-(\mathbf{v})\right)^2}, \qquad (18)$$

are local scaling factors still depending on the particular choices of d and f. Here f' denotes the derivative $\frac{\partial f}{\partial\mu}$. Again, α is the general learning rates as before. Hence, the factor $\alpha\xi_\mu^\pm(\mathbf{v},f,d)$ in (17) can be seen as a *localized* learning rate for $\mathbf{v}$. Further, the derivative $\frac{\partial d(\mathbf{v},\mathbf{w}^\pm)}{\partial\mathbf{w}^\pm}$ becomes a vector shift $-2(\mathbf{v}-\mathbf{w}^\pm)$ in case of the squared Euclidean distance d_E such that (17) becomes the repulsing and attraction shift as known from LVQ2.1 [15].

It turns out that GLVQ belongs to the margin optimizing classifiers as the popular SVM. In contrast to SVM, which maximizes the separation margin, GLVQ optimizes the local hypothesis margin

$$m_{lh}(\mathbf{v}) = \left|d^-(\mathbf{v}) - d^+(\mathbf{v})\right|,$$

corresponding to the global hypothesis margin

$$m_h = \max_{j,k}\left\{d(\mathbf{w}_j,\mathbf{w}_k)\,|\,c(\mathbf{w}_j)\neq c(\mathbf{w}_k)\right\},$$

and being a lower bound of the separation margin [31].

A probabilistic approach of LVQ was proposed in [32]. For this purpose, the log-likelihood ratio using a Gaussian mixture model is considered and a SGDL scheme is provided. This so-called Robust Soft LVQ (RSLVQ) delivers frequently very stable solutions. Alternatively, a Gaussian mixture model with direct SGDL was investigated in [33]. The resulting soft nearest neighbor classification approach can be seen as an annealed version of LVQ. An information theoretic GLVQ variant based on cross entropy optimization was introduced in [34], however, showing stability problems in convergence.

One of the major problems in GLVQ learning concerns the sensitivity with respect to initialization of the prototypes as well as the the avoidance of dead units (prototypes). This problem can be tackled by incorporation of neighborhood cooperativeness as known from SOM and NG [35]. An alternative possibility was provided in [36], proposing a parametrized distance measure based on the harmonic mean of all prototype distances to the given input, which smoothly exchanges to the standard distance in dependence on the parameter.

3 Basic GLVQ - a starting tool for task specific classifier design

The basic GLVQ model can be seen as an ideal starting point to generate adequate classifier models depending on the specific task. In this chapter we will review the most challenging variants developed so far. We will briefly describe the basic ideas of these GLVQ variants but refer for details to the specific publications.

3.1 Class typical prototypes versus class border-sensitive LVQ variants

One of the main aspects contributing to the popularity of LVQ methods is the intuitive assumption that the prototypes are localized within the class distribution, i.e. the prototypes are representatives of their classes. Although this behavior is frequently observed, it is not ensured by the GLVQ algorithm mathematically, see Fig. (1).

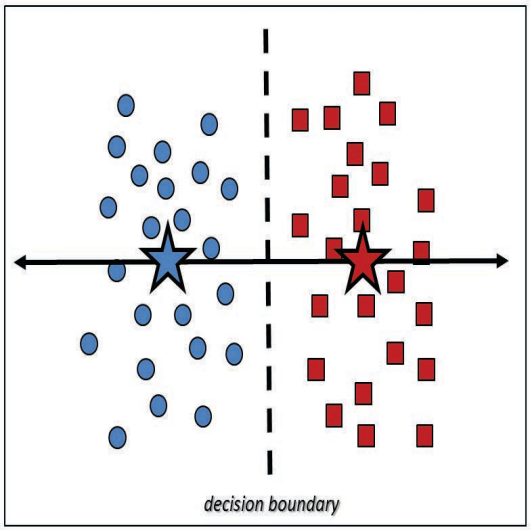

Figure 1. Representative versus non-representative prototypes: The prototypes (stars) can be moved following the horizontal arrow without any change of the classification accuracy provided their distance to the decision boundary is equal.

The respective problem of generative (representative) models versus classification property was generally discussed for vector quantization [9, 37, 38, 39]. In consequence, if a generative GLVQ models is strictly demanded, one has to add a respective penalty term to the cost function according to

$$\begin{aligned} &_{GLVQ-generative}(V,W,f,d) = \\ &E_{GLVQ}(V,W,f,d) + \vartheta \sum_{\mathbf{v}\in V} d^s(\mathbf{v}), \end{aligned} \tag{19}$$

forcing this behavior scaled by the parameter $\vartheta > 0$ [38].

Otherwise, if it is desirable to obtain prototypes describing the class borders more precisely, there are at least two possibilities: The first way is again to add a cost term penalizing large distances between $\mathbf{w}^+$ and $\mathbf{w}^-$, i.e.

$$\begin{aligned} &_{GLVQ-border\,localized}(V,W,f,d) = \\ &E_{GLVQ}(V,W,f,d) + \gamma \sum_{\mathbf{v}\in V} d(\mathbf{w}^+(\mathbf{v}),\mathbf{w}^-(\mathbf{v})), \end{aligned} \tag{20}$$

compelling the prototypes to be localized at the class borders controlled by the parameter $\gamma > 0$ [39]. If the transfer function f is specified as the sigmoid function f_θ from (14), the parameter θ can be used to control the border-sensitivity of the GLVQ. Because the derivative f'_θ is involved into the prototype updates (17) via the scaling factors $\xi^{\pm}_{\mu}(\mathbf{v},f,d)$ from (18). Particularly, we have the derivative

$$f'_\theta(\mu(\mathbf{v})) = \frac{f_\theta(\mu(\mathbf{v}))}{\theta} \cdot (1 - f_\theta(\mu(\mathbf{v}))), \tag{21}$$

determining the strength of $\xi^{\pm}_{\mu}(\mathbf{v},f,d)$. A significant prototype update only takes place for a small range of the classifier values μ in (11) depending on the parameter θ corresponding to the so-called *active set*

$$\hat{\Xi} = \left\{ \mathbf{v} \in V | \mu(\mathbf{v}) \in \left[-\frac{1-\mu_\theta}{1+\mu_\theta}, \frac{1-\mu_\theta}{1+\mu_\theta} \right] \right\}, \tag{22}$$

with μ_θ chosen such that $f'_\theta(\mu) \approx 0$ is valid for $\mu \in \Xi = V \setminus \hat{\Xi}$, see Fig. 2.

In consequence, the prototypes are responsible during learning only for the active set $\hat{\Xi}$, which is located alongside the class borders. Therefore, they are attracted by these regions and become border-sensitive. We emphasize at this point that the prototypes do not necessarily located near the class borders in difference to the previously discussed border-localized GLVQ.

Thus, both class border-responsible variants can bee seen as alternatives to SVMs, which identify data samples determining the class borders as support vectors.

3.2 Beyond the Euclidean world – GLVQ with non-standard dissimilarities and kernel metrics

Most of the models in machine learning and pattern recognition including the original LVQ and GLVQ are originally introduced based on the Euclidean distance or are immediate side-products like the RBF-kernel

$$\kappa_\sigma(\mathbf{v},\mathbf{w}_k) = \exp\left(-\frac{d_E(\mathbf{v},\mathbf{w}_k)}{2\sigma^2} \right). \tag{23}$$

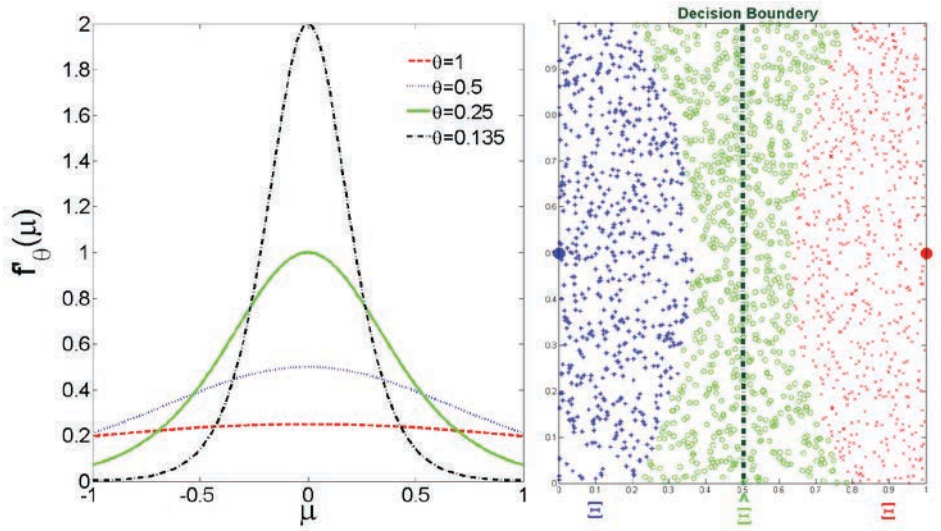

Figure 2. **left)** derivatives $f'_\theta(\mu)$ for different θ-values; **right)** Visualization of the active set $\hat{\Xi}$ (green points) for a simple example. The prototypes are the big dots. Figure taken from [39].

Yet, the Euclidean metric is not adequate for many problems. For example, data describing densities might be better distinguished using divergences or respective mutual information. In biology, frequently data are compared by correlations. More generally, one can think in terms of general dissimilarities and similarities [40, 41]. Thus it is necessary to adapt classifier models also for those scenarios.

As we learned before, the requirement regarding the dissimilarity applied in GLVQ is differentiability. Hence, the (squared) Euclidean distance can simply replaced by an arbitrary dissimilarity or similarity measure, whereby in the latter case the gradient descent has to be replaced by a gradient ascent. Several investigation show that this strategy can be successfully applied in GLVQ. One of the easiest possibilities to leave the Euclidean world is to apply d_p from (6) directly with $p \neq 2$. This requires approximations of the absolute value function to ensure the differentiability [42]. In case of complex-valued data the differentiability conditions become at least critical [43]. Adequate processing requires a precise interpretation of the differentiability, the most convenient in context of GLVQ seems to be the so-called Wirtinger calculus [44]. Divergences for density and histogram data are considered in [45, 46], metrics for functional data were reported in [47, 48, 49]. Correlation based GLVQ as preferred frequently in biological application was introduced in [50] using the differentiability of the Pearson correlation. Recent developments include tangent metrics to deal with invariances of data regarding classification, e.g. rotations of objects in images [51, 52].

One of the most important alternatives to the Euclidean space in case of difficult classification tasks is an arbitrary Hilbert space $\mathcal{H}$ with potentially infinite dimension providing a greater flexibility for model adaptation. If the Hilbert space is related to a kernel κ constituting the respective inner product, SVMs use the implicit data mapping ϕ_κ to $\mathcal{H}$ to to exploit this flexibility [53, 54]. Yet, several attempts were made to incorporate the kernel approach also into LVQ schemes. First approaches use finite approximations in the infinite Hilbert space [55, 56]. A more elegant way is to replace the Euclidean distance directly by the kernel distance d_κ generated from the kernel κ. If the kernel distance is $d_\kappa(\mathbf{v}, \mathbf{w}_k)$ is differentiable like the RBF kernel (23), we can immediately plugin this into GLVQ obtaining a kernel variant, which works exactly in the same Hilbert space [57]. Obviously, this trick could also be applied to RSLVQ.[1]

Yet, the properties of general (dis-) similarities may cause surprising learning effects due to unexpected behavior. The most prominent example is that kernels as inner products in a Hilbert space do not necessarily be similarity measures. For a respective discussion we refer to [40, 59].

[1] Direct application of kernels instead of kernel distances was investigated for RSLVQ [58].

If differentiability of the given data dissimilarity is not valid, embedding techniques could offer a possibility to handle the data [40]. Euclidean embedding for GLVQ assumes the prototypes to be linear combinations of the data and the prototype learning takes places as the adaptation of the respective coefficients [60]. If embedding is not adequate, a median variant of GLVQ can be applied, which restricts the prototypes to be data samples and uses a generalized expectation maximization scheme for learning [61].

3.3 Relevance learning and related variants

Frequently, the classification is not based on all available data features. Only a few contribute to the decision. Thus a feature weighting according by means of the weighted Euclidean distance

$$d_\lambda(\mathbf{v},\mathbf{w}) = \sum_{i=1}^{n} \lambda_i (v_i - w_i), \tag{24}$$

with non-negative weights λ_i as dissimilarity measure would be more appropriate. However, generally it is not known in advance, which data feature are relevant. The classification task depending feature weighting can be realized within the GLVQ framework yielding the generalized relevance LVQ (GRLVQ,[62]) with the cost function $E_{GRLVQ}(V,W,f,d_\lambda)$ now depending on d_λ compared to GLVQ. The task dependent adjustment of the weights λ_i are obtained as SGDL of $E_{GRLVQ}(V,W,f,d_\lambda)$ with respect to λ_i, which takes place at the same time as the prototype adaptation. This weight adaptation is denoted as *relevance learning*. After training high relevance weights λ_i indicate high feature importance for the classification. Thus the relevance profile vector λ delivers information about the necessary data features for class discrimination. It turns out that the GRLVQ still optimizes the hypothesis margin with generalization bound deducible similar to SVM [63]. If the data are functional data, i.e. the data vectors are discrete realizations of function and, therefore, n becomes frequently large leading to a slowed convergence speed of GRLVQ during training. Yet, the relevance learning can be accelerated taking into account this functional behavior [47, 64].

If also linear combination of the features are of interest, the matrix GLVQ (GMLVQ, [65]) is adequate taking the quadratic form

$$d_\Omega(\mathbf{v},\mathbf{w}) = (\mathbf{v}-\mathbf{w})^T \Omega^T \Omega (\mathbf{v}-\mathbf{w}), \tag{25}$$

with $\Omega \in \mathbb{R}^{m\times n}$ as dissimilarity measure, whereby in the standard method $m = n$ is valid. Thus the matrix $\Lambda = \Omega^T\Omega \in \mathbb{R}^{n\times n}$ determines the linear correlations between the data features. Note that d_Ω remains a squared Euclidean distance according to the equivalence

$$d_\Omega(\mathbf{v},\mathbf{w}) = (\Omega\mathbf{v} - \Omega\mathbf{w})^2, \tag{26}$$

where Ω can be interpreted as projection matrix. If one considers the respective GLVQ cost function $E_{GMLVQ}(V,W,f,d_\Omega)$ taking d_Ω as dissimilarity in GLVQ and adapts the matrix entries Ω_{ij} via SGDL, Λ yields a so-called *classification correlation matrix* indicating those linear correlation, which contribute to class discrimination. Usually, regularization techniques and have to accompany the matrix adaptation to achieve stable behavior [66, 67]. GRLVQ is obtained restricting GMLVQ to diagonal matrices Λ. If $m < n$, a limited rank version is installed, which can be used for class separating data visualization [68]: If $m \leq 3$, GMLVQ optimizes the approximated classification error according to both the localization of prototypes as well as the projection matrix Ω.

If heterogeneous or structured data with several components $[\mathbf{v}]_l$ have to be processed, a single dissimilarity measure is not sufficient. Here combined measures $d_{comb}(\mathbf{v},\mathbf{w}) = \sum_l \gamma_l d_l([\mathbf{v}]_l,[\mathbf{w}]_l)$ have to replace a single one, whereby each of the submeasures d_l specifically designed for the respective data component compared with the prototype component $[\mathbf{w}]_l$ and the non-negative weights γ_l describe the influence [69, 70]. This approach is comparable to deep learning neural networks for structured data with specialized neurons [71]. Obviously, the relevance learning technique can be transferred immediately to this situation to optimize the influence of the components via the component weights γ_l. Another, possibility is to apply different dissimilarities to the whole data vector and compare them regarding their performance for class discrimination [72, 73].

Otherwise, if the data classes show invariances regarding data transformations the classifier should reflect this property, i.e. the classifier should recognize transformed data adequately. One possibility would be to extract respective invariant features as it

is frequently done in pattern recognition and image processing [74, 75]. Another possibility is to apply tangent metrics, which handle the data as a certain point of a manifold describing the possible transformations [76, 77]. The crucial point is, which transformations are assumed to be in play. Recently, GLVQ was extended also to deal with this problem. More specifically, GLVQ is provided wih an adaptive tangent metric (GTLVQ) allowing to determine the respective invariances/transformations automatically during classification learning following the same principle as relevance learning [51, 52].

Related to this subject is the classic transfer or representation learning [78, 79]. Here, after initial training of the network new data become available. However, these data are slightly different from the initial data, i.e. they can be seen as transformed data. to avoid complete new learning of the transfer data (and maybe destroying the knowledge of the already trained model), the transfer data should be processed using the already acquired information of the initial learning. First attempts for GLVQ are presented in [80]. Yet, it is obvious that this problem could be also tackled by GTLVQ learning first the prototypes by usual GLVQ for th initial data followed by the tangent learning according to the transfer data.

So far we discussed only global metrics/dissimilarities, i.e. the dissimilarities are equally applied for all prototypes. Localized variants are obtained if each prototype is equipped with its own dissimilarity. These localized GLVQ models offer further improvements and flexibility making them comparable to advanced methods for SVM and deep architectures.

3.4 Beyond the accuracy - Optimization of other statistical classification measures

As we pointed out previously, the cost function (15) of GLVQ approximates the classification error (11). Yet, the classification is not appropriate in case of imbalanced data [81, 82]. Here other statistical measures are demanded to detect an accurate classification property adequately. For binary classification, frequently sensitivity (recall) ρ, precision π, and specificity ς are better suited to assess those situations [83, 84], which all are calculated from the contingency table analysis based on the true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). The F_β-measure

$$F_\beta = \frac{\left(1+\beta^2\right)\cdot\pi\cdot\rho}{\beta^2\cdot\pi+\rho}, \tag{27}$$

developed by C.J. van Rijsbergen is well accepted in engineering [85]. For the common choice $\beta = 1$, F_β is the fraction of the harmonic and the arithmetic mean of precision and recall, i.e. β controls the influence of both values.

Otherwise, we remind that detecting (exact) classification error is nothing else to count and to collect them in FP and FN. In GLVQ, the classification error $CE = \frac{FP+FN}{N}$ this is approximated by the cost function (15). Hence, one can apply this approximation technique also to count all the contingency table quantities TP, TN, FP, and FN and combine them as required. Respective GLVQ variants are provided in [86] for the continuous as well as the median variant.

Alternatively, the receiver-operator-characteristic (ROC) is a advanced measure for classifier comparison in machine learning [87, 88]. Each classifier generates a TP-rate and a FP-rate and, thus, is a single point in the ROC-space, see Fig. (3)(left). The analysis of the ROC-curve for parametrized classifiers is an established method to select optimum parameter configurations or to compare the classifier performances [82, 89].

The latter problem is tackled investigating the area under the ROC-curve (AUROC) [91]. Here, the ROC-curve is the parameter depended curve consisting of the respective TP-rate-FP-rate-pairs, see Fig. (3)(right). The best parametrized classifier is obtained for maximum AUROC.

Several attempts were made to include this scheme into neural networks and SVM [92, 93]. Yet, these approaches do not directly optimize the AUROC.

For GLVQ, a respective regime can be directly obtained using the parameter dependent classifier function

$$\mu_d^W(\mathbf{v},\gamma) = \mu_d^W(\mathbf{v}) - \gamma, \tag{28}$$

instead of (11). However, prototype optimization for this model requires to take the AUROC as cost function for GLVQ. Fortunately, this can easily be realized using structured input consisting of pairs of data vectors corresponding to both classes. For details we refer to [94, 90].

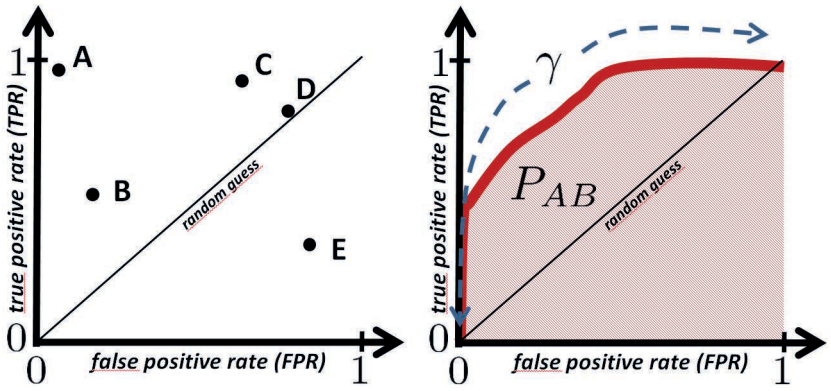

Figure 3. Visualization of the ROC space with classifier performances. **left:** 5 classifiers are displayed according to there performances. A- nearly perfect classifier, B - 'conservative' classifier, C - 'liberal' classifier, D - random guess classifier, E - worse than random guess classifier (adapted from [82]); **right:** ROC curve for a classifier with continuous discriminant function and parameter γ. Different γ-values correspond to different classifier performances generating the ROC-curve. The area under the ROC-curve (AUROC) is equal to the probability P_{AB} that a classifier will rank a randomly chosen A-instance higher than a randomly chosen B-instance. (according to [90])

Summarizing this sub-section, the basic GLVQ-scheme can easily be adapted to more sophisticated statistical quality measures than the approximated classification error. Thus, the GLVQ-approach provide a great flexibility for user specific requirements regarding classification assessment.

3.5 Reject or classify - secure classification

Classification decisions by machine learning systems are always afflicted with uncertainty. According to [95], one can distinguish aleatoric and epistemic uncertainty in the classifier. In the context of LVQ, aleatoric uncertainty is due to randomness in data generating scheme whereas epimistic uncertainty is related to the lack of knowledge. The latter case occurs if we observe overlapping class distributions in the training data, i.e. ambiguous regions near the classs borders whereas an example for aleatoric uncertainty are outliers in data [96, 97], see Fig. 4.

Several heuristic approaches were proposed to deal with these problems [98, 99, 100] or to treat it as a probabilistic approach [101, 102]. In LVQ systems usually reject options are applied after model training [103, 104]. An attempt to incorporate reject options into the SVM model was provided in [105], which is based on geometric considerations. Yet, geometric approaches depend on the utilized dissimilarity measure.

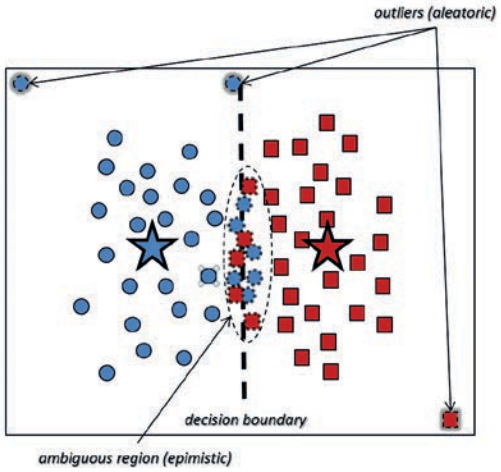

Figure 4. Visualization of aleatoric and epimistic reject regions in the data space.

Recently, GLVQ was also adapted to deal with reject options already during learning. For this purpose, the theoretical framework for cost-based classification introduced by Chow was used [106]. This framework assigns to each classification decision as well as to the reject decision cost. Based on the Bayes theory a cost function is provided reflecting the the expected costs for errors C_e, rejects C_r, and correct decisions C_c, whereby an proper scaling always ensures $C_c = 1$ [107]. Accordingly, one can define outlier costs C_o in outlier detection for the aleatoric uncertainty [108]. Feeding these advisements to the GLVQ we obtain

$$_{GLVQ-rejectoption}(V,W,f,d,C_e,C_r,C_o) = \sum_v E_e(v)+E_r(v)+E_o(v), \tag{29}$$

with the local classification error

$$E_e(\mathbf{v}) = C_e f\left(\mu_d^W(\mathbf{v})+\varepsilon_{ep}\right), \tag{30}$$

depending on adaptive epimistic threshold ε_{ep} and the classification error costs C_e, the local classification reject error

$$E_r(\mathbf{v}) = (C_e - C_r) f\left(\mu_d^W(\mathbf{v})-\varepsilon_{ep}\right), \tag{31}$$

involving additionally the reject costs C_r, and the outlier reject error

$$E_e(\mathbf{v}) = C_o\left(d\left(\mathbf{v},\mathbf{w}_{s(\mathbf{v})}\right)-\varepsilon_o\right), \tag{32}$$

to be a function of the aleatoric oulier threshold ε_o [108, 109, 110]. Obviously, $E_{GLVQ-reject option}$ can be optimized by SGDL in complete analogy to standard GLVQ yielding similar update rules for the prototypes. Moreover, the thresholds ε_{ep} and ε_o, which determine geometric reject regions, can be also adapted by SGDL such that optimum values are ascertained depending on the pre-defined costs.

An alternative but related approach to deal with uncertain classifications is to equip each classification decision with a certainty probability [111]. For LVQ systems this was realized based on the classifier function $\mu_d^W(\mathbf{v})$ and, therefore, also being a geometric approach [112]. However, the learning of GLVQ is not changed in this approach, because the so-called conformal prediction probability is only calculated for a classification decision in the recall phase when an unknown objects has to be classified but is not influencing any GLVQ learning scheme.

4 Conclusion

In this overview article we summarized recent developments in prototype based learning vector quantization for classification learning. Particularly, we show that the basic but theoretically justified GLVQ model can be easily adapted to diverse application requirements ranging from data specific dissimilarity measures and different classification success assessment to problems of certainty and secure classification. Most of these adaptations are intuitively comprehensible although mathematically verified and allow good interpretability. Thus GLVQ with variants can be seen as a basic module with task specific modifications keeping the basic principles offering a valuable alternative to SVM or deep learning techniques, which frequently are difficult to interpret and require advanced theoretical knowledge for correct use.

Another important advantage of LVQ models is the pre-defined model complexity determined by the number of prototype in advance. This feature becomes important, if only restricted resources are available like in many real-world sensor systems. Here limited memory as well as fast calculations may play a key role for a successful application. As previously discussed advanced driving assistance systems on car can neither use expansive pre-trained subnetworks as it is frequently the case in high performance deep learning architectures [113, 114], nor a huge number of support vectors provided by a SVM requiring expansive calculations during the application phase. For those applications smart models, like LVQ networks with only a few prototypes, with maybe slightly lower but still acceptable performance are preferred.

As already mentioned in the introduction, we did not explicitly addressed in detail the problem of sensitivity of LVQ networks with respect to initialization. However, this difficulty can be solved involving the neighborhood learning idea from neural maps like SOM or NG into GLVQ (and its variants) [35] or applying relaxing techniques [36]. Thus we refer the reader to these investigations well-known in neural computation and machine learning. Yet, neighborhood cooperativeness between the processing units (here prototypes) seems to be one of the best accelerating techniques for learning as it is was adopted from neural map learning in cortical brain areas [115].

Understandably, this systematic LVQ overview can only serve as a starting point for further research and applications. Neither, this contribution is comprehensive at all nor provides the here chosen systematic approach the only point of view for prototype based classification in general and learning vector quantization in particular. It is rather a good survey of the state of the art for more detailed investigations when searching for a classification algo-

rithm with special user specific requirements based on a few simple and intuitive principles.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. Nature, 521:436–444, May 2015.

[2] P.J. Werbos. Beyond Regression: New Tools for Prediction and Analysis in the Behavorial Sciences. PhD thesis, Havard University, Cambridge, MA., 1974.

[3] G. Cybenko. Approximations by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems, 2(4): 303–314, 1989.

[4] T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning. Springer Verlag, Heidelberg-Berlin, 2001.

[5] Y. Bengio. Learning deep architectures for AI. Foundations and Trends in Machine Learning, 2(1):1–127, 2009.

[6] Simon Haykin. Neural Networks - A Comprehensive Foundation. IEEE Press, New York, 1994.

[7] C.M. Bishop. Pattern Recognition and Machine Learning. Springer, 2006.

[8] R.O. Duda and P.E. Hart. Pattern Classification and Scene Analysis. Wiley, New York, 1973.

[9] K.L. Oehler and R.M. Gray. Combining image compressing and classification using vector quantization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(5):461–473, 1995.

[10] M. Biehl, B. Hammer, and T. Villmann. Prototype-based models in machine learning. Wiley Interdisciplinary Reviews: Cognitive Science, 7(2):92–111, 2016.

[11] P. L. Zador. Asymptotic quantization error of continuous signals and the quantization dimension. IEEE Transaction on Information Theory, IT-28:149–159, 1982.

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

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

[14] J.C. Principe. Information Theoretic Learning. Springer, Heidelberg, 2010.

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

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

[17] B. Schlkopf and A. Smola. Learning with Kernels. MIT Press, Cambridge, 2002.

[18] Teuvo Kohonen. Learning vector quantization for pattern recognition. Report TKK-F-A601, Helsinki University of Technology, Espoo, Finland, 1986.

[19] Teuvo Kohonen. Learning Vector Quantization. Neural Networks, 1(Supplement 1):303, 1988.

[20] Teuvo Kohonen. Improved versions of Learning Vector Quantization. In Proc. IJCNN-90, International Joint Conference on Neural Networks, San Diego, volume I, pages 545–550, Piscataway, NJ, 1990. IEEE Service Center.

[21] D. Nova and P.A. Estévez. A review of learning vector quantization classifiers. Neural Computation and Applications, 25(511–524), 2013.

[22] M. Kaden, M. Lange, D. Nebel, M. Riedel, T. Geweniger, and T. Villmann. Aspects in classification learning - Review of recent developments in Learning Vector Quantization. Foundations of Computing and Decision Sciences, 39(2):79–105, 2014.

[23] B. Fritzke. The LBG-U method for vector quantization - an improvement over LBG inspired from neural networks. Neural Processing Letters, 5(1):35–45, 1997.

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

[25] F. Hamker. Life-long learning cell structures – continuously learning without catastrophic interference. Neural Networks, 14:551–573, 2001.

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

[27] H.J. Kushner and D.S. Clark. Stochastic Approximation Methods for Constrained and Unconstrained Systems. Springer-Verlag, New York, 1978.

[28] S. Graf and H. Luschgy. Foundations of Quantization for Probability Distributions, volume 1730 of Lect. Notes in Mathematics. Springer, Berlin, 2000.

[29] G. Voronoi. Nouvelles aoolications des parametres la theorie des formes quadratiques. deuxime mmorie: Recherches sur les paralllodres primitifs. J. reine angew. Math., 134:198–287, 1908.

[30] 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.

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

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

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

[34] A. Boubezoul, S. Paris, and M. Ouladsine. Application of the cross entropy method to the GLVQ algorithm. Pattern Recognition, 41:3173–3178, 2008.

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

[36] A.K. Qin and P.N. Suganthan. Initialization insensitive LVQ algorithm based on cost-function adaptation. Pattern Recognition, 38:773–776, 2004.

[37] Keren O. Perlmutter, Sharon M. Perlmutter, Robert M. Gray, Richard A. Olshen, and Karen L. Oehler. Bayes risk weighted vector quantization with posterior estimation for image compression and classification. IEEE Trans. on Image Processing, 5(2):347–360, February 1996.

[38] B. Hammer, D. Nebel, M. Riedel, and T. Villmann. Generative versus discriminative prototype based classification. In T. Villmann, F.-M. Schleif, M. Kaden, and M. Lange, editors, Advances in Self-Organizing Maps and Learning Vector Quantization: Proceedings of 10th International Workshop WSOM 2014, Mittweida, volume 295 of Advances in Intelligent Systems and Computing, pages 123–132, Berlin, 2014. Springer.

[39] M. Kaden, M. Riedel, W. Hermann, and T. Villmann. Border-sensitive learning in generalized learning vector quantization: an alternative to support vector machines. Soft Computing, 19(9):2423–2434, 2015.

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

[41] T. Villmann, M. Kaden, D. Nebel, and A. Bohnsack. Data similarities, dissimilarities and types of inner products - a mathematical characterization in the context of machine learning. Machine Learning Reports, 9(MLR-04-2015):19–29, 2015. ISSN:1865-3960, http://www.techfak.uni-bielefeld.de/~fschleif/mlr/mlr_04_2015.pdf.

[42] M. Lange, D. Zühlke, O. Holz, and T. Villmann. Applications of l_p-norms and their smooth approximations for gradient based learning vector quantization. In M. Verleysen, editor, Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2014), pages 271–276, Louvain-La-Neuve, Belgium, 2014. i6doc.com.

[43] K. Bunte, F.-M. Schleif, and M. Biehl. Adaptive learning for complex-valued data. In M. Verleysen, editor, Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2012), pages 381–386, Louvain-La-Neuve, Belgium, 2012. i6doc.com.

[44] M. Gay, M. Kaden, M. Biehl, A. Lampe, and T. Villmann. Complex variants of GLVQ based on Wirtingers calculus. In E. Merényi, M.J. Mendenhall, and P. O'Driscoll, editors, Advances in Self-Organizing Maps and Learning Vector Quantization: Proceedings of 11th International Workshop WSOM 2016, volume 428 of Advances in Intelligent Systems and Computing, pages 293–303, Berlin-Heidelberg, 2016. Springer.

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

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

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

[48] F. Rossi, N. Delannay, B. Conan-Gueza, and M. Verleysen. Representation of functional data in neural networks. Neurocomputing, 64:183–210, 2005.

[49] F. Melchert, U. Seiffert, and M. Biehl. Functional representation of prototypes in lvq and relevance learning. In E. Merényi, M.J. Mendenhall, and P. O'Driscoll, editors, Advances in Self-Organizing Maps and Learning Vector Quantization: Proceedings of 11th International Workshop WSOM 2016, volume 428 of Advances in Intelligent Systems

and Computing, pages 317–327, Berlin-Heidelberg, 2016. Springer.

[50] 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.

[51] S. Saralajew and T. Villmann. Adaptive tangent metrics in generalized learning vector quantization for transformation and distortion invariant classification learning. In Proceedings of the International Joint Conference on Neural networks (IJCNN) , Vancover, pages 2672–2679. IEEE Computer Society Press, 2016.

[52] S. Saralajew, D. Nebel, and T. Villmann. Adaptive Hausdorff distances and tangent distance adaptation for transformation invariant classification learning. In A. Hirose, editor, Proceedings of the International Conference on Neural Information Processing (ICONIP) , Kyoto, volume 9949 of LNCS, pages 362–371. Springer, 2016.

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

[54] I. Steinwart and A. Christmann. Support Vector Machines. Information Science and Statistics. Springer Verlag, Berlin-Heidelberg, 2008.

[55] A.K. Qin and P.N. Suganthan. A novel kernel prototype-based learning algorithm. In Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04), volume 4, pages 621–624, 2004.

[56] F.-M. Schleif, T. Villmann, B. Hammer, and P. Schneider. Efficient kernelized prototype based classification. International Journal of Neural Systems, 21(6):443–457, 2011.

[57] T. Villmann, S. Haase, and M. Kaden. Kernelized vector quantization in gradient-descent learning. Neurocomputing, 147:83–95, 2015.

[58] D. Hofmann, A. Gisbrecht, and B. Hammer. Efficient approximations of robust soft learning vector quantization for non-vectorial data. Neurocomputing, 147:96–106, 2015.

[59] D. Nebel, M. Kaden, A. Bohnsack, and T. Villmann. Types of (dis-)similarities and adaptive mixtures thereof for improved classification learning. Neurocomputing, page in press, 2017.

[60] B. Hammer, D. Hofmann, F.-M. Schleif, and X. Zhu. Learning vector quantization for (dis-)similarities. Neurocomputing, 131:43–51, 2014.

[61] D. Nebel, B. Hammer, K. Frohberg, and T. Villmann. Median variants of learning vector quantization for learning of dissimilarity data. Neurocomputing, 169:295–305, 2015.

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

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

[64] T. Villmann, M. Kästner, D. Nebel, and M. Riedel. Lateral enhancement in adaptative metric learning for functional data. Neurocomputing, 131:23–31, 2014.

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

[66] P. Schneider, K. Bunte, H. Stiekema, B. Hammer, T. Villmann, and Michael Biehl. Regularization in matrix relevance learning. IEEE Transactions on Neural Networks, 21(5):831–840, 2010.

[67] M. Biehl, B. Hammer, F.-M. Schleif, P. Schneider, and T. Villmann. Stationarity of matrix relevance LVQ. In Proc. of the International Joint Conference on Neural Networks 2015 (IJCNN), pages 1–8, Los Alamitos, 2015. IEEE Computer Society Press.

[68] 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.

[69] E. Mwebaze, G. Bearda, M. Biehl, and D. Zühlke. Combining dissimilarity measures for prototype-based classification. In M. Verleysen, editor, Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2015), pages 31–36, Louvain-La-Neuve, Belgium, 2015. i6doc.com.

[70] D. Zühlke, F.-M. Schleif, T. Geweniger, S. Haase, and T. Villmann. Learning vector quantization for heterogeneous structured data. In M. Verleysen, editor, Proc. of European Symposium on Artificial Neural Networks (ESANN'2010), pages 271–276, Evere, Belgium, 2010. d-side publications.

[71] J. Schmidhuber. Deep learning in neural networks: An overview. Neural Networks, 61:85–117, 2015.

[72] U. Knauer, A. Backhaus, and U. Seiffert. Beyond standard metrics - on the selection and combination of distance metrics for an improved classification of hyperspectral data. In T. Villmann, F.-M. Schleif, M. Kaden, and M. Lange, editors, Advances in Self-Organizing Maps and Learning Vector Quantization:

Proceedings of 10th International Workshop WSOM 2014, Mittweida, volume 295 of Advances in Intelligent Systems and Computing, pages 167–177, Berlin, 2014. Springer.

[73] M. Kaden, D. Nebel, and T. Villmann. Adaptive dissimilarity weighting for prototype-based classification optimizing mixtures of dissimilarities. In M. Verleysen, editor, Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2016), pages 135–140, Louvain-La-Neuve, Belgium, 2016. i6doc.com.

[74] D.G. Lowe. Object recognition from local scale-invariant features. In The Proceedings of the Seventh IEEE International Conference on Computer Vision, volume 2, pages 1150–1157, 1999.

[75] D.G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2):91–110, 2004.

[76] P. Simard, Y. LeCun, and J.S. Denker. Efficient pattern recognition using a new transformation distance. In S.J. Hanson, J.D. Cowan, and C.L. Giles, editors, Advances in Neural Information Processing Systems 5, pages 50–58. Morgan-Kaufmann, 1993.

[77] T. Hastie, P. Simard, and E. Säckinger. Learning prototype models for tangent distance. In G. Tesauro, D.S. Touretzky, and T.K. Leen, editors, Advances in Neural Information Processing Systems 7, pages 999–1006. MIT Press, 1995.

[78] S.J. Pan and Q. Yang. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10):13–451359, 2010.

[79] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8):1798–1828, 2013.

[80] C. Prahm, B. Paassen, A. Schulz, B. Hammer, and O. Aszmann. Transfer learning for rapid recalibration of a myoelectric prosthesis after electrode shift. In J. Ibanez, J. Gonzales-Vargas, J.M. Azorin, M.Akay, and J.L. Pons, editors, Proceedings of the 3rd International Conference on NeuroRehabilitation (ICNR2016), volume 15 of Biosystems and Biorobotics, pages 153–157. Springer, 2016.

[81] Y.Tang, Y.Q. Zangh, N.V. Chawla, and S. Krasser. SVMs modeling for highly imbalanced classification. IEEE Transactions on Systems Man and Cybernetics, Part B, 39(1):281–288, 2009.

[82] T. Fawcett. An introduction to ROC analysis. Pattern Recognition Letters, 27:861–874, 2006.

[83] P. Baldi, S. Brunak, Y. Chauvin, and C. Andersen H. Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics, 16(5):412–424, 2000.

[84] L. Sachs. Angewandte Statistik. Springer Verlag, 7-th edition, 1992.

[85] C.J. Rijsbergen. Information Retrieval. Butterworths, London, 2nd edition edition, 1979.

[86] M. Kaden, W. Hermann, and T. Villmann. Optimization of general statistical accuracy measures for classification based on learning vector quantization. In M. Verleysen, editor, Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2014), pages 47–52, Louvain-La-Neuve, Belgium, 2014. i6doc.com.

[87] A.P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition, 30(7):1149–1155, 1997.

[88] J. Keilwagen, I. Grosse, and J. Grau. Area under precision-recall curves for weighted and unweighted data. PLOS|ONE, 9(3 / e92209):1–13, 2014.

[89] S. Vanderlooy and E. Hüllermeier. A critical analysis of variants of the AUC. Machine Learning, 72:247–262, 2008.

[90] T. Villmann, M. Kaden, W. Hermann, and M. Biehl. Learning vector quantization classifiers for ROC-optimization. Computational Statistics, 2016.

[91] J.A. Hanley and B.J. McNeil. The meaning and use of the area under a receiver operating characteristic. Radiology, 143:29–36, 1982.

[92] U. Brefeld and T. Scheffer. AUC maximizing support vector learning. In Proceedings of ICML 2005 workshop on ROC Analysis in Machine Learning, pages 377–384, 2005.

[93] T. Calders and S. Jaroszewicz. Efficient AUC optimization for classification. In J.N. Kok, J. Koronacki, R. Lopez de Mantaras, S. Matwin, D. Mladenic, and A. Skowron, editors, Knowledge Discovery in Databases: PKDD 2007, volume 4702 of LNCS, pages 42–53. Springer-Verlag, 2007.

[94] M. Biehl, M. Kaden, P. Stürmer, and T. Villmann. ROC-optimization and statistical quality measures in learning vector quantization classifiers. Machine Learning Reports, 8(MLR-01-2014):23–34, 2014. ISSN:1865-3960, http://www.techfak.uni-bielefeld.de/~fschleif/mlr/mlr_01_2014.pdf.

[95] R. Senge, S. Bösner, Dembczyński K, J. Haasenritter, O. Hirsch, N. Donner-Banzhoff, and E. Hüllermeier. Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty. Information Sciences, 255:16–29, 2014.

[96] A. Vailaya and A.K. Jain. Reject option for VQ-based Bayesian classification. In International Conference on Pattern Recognition (ICPR), pages 2048–2051, 2000.

[97] L. Fischer, B. Hammer, and H. Wersing. Efficient rejection strategies for prototype-based classification. Neurocomputing, 169:334–342, 2015.

[98] G. Fumera, F. Roli, and G. Giacinto. Reject option with multiple thresholds. Pattern Recognition, 33(12):2099–2101, 2000.

[99] I. Pillai, G. Fumera, and F. Roli. Multi-label classification with a reject option. Pattern Recognition, 46:2256–2266, 2013.

[100] R. Herbei and M.H. Wegkamp. Classification with reject option. The Canadian Journal of Statistics, 34(4):709–721, 2006.

[101] P. L. Bartlett and M.H. Wegkamp. Classification with a reject option using a hinge loss. Journal of Machine Learning Research, 9:1823–1840, 2008.

[102] M. Yuan and M.H. Wegkamp. Classification methods with reject option based on convex risk minimization. Journal of Machine Learning Research, 11:111–130, 2010.

[103] L.P. Cordella, C. deStefano, C. Sansone, and M. Vento. An adaptive reject option for LVQ classifiers. In C. Braccini, L. deFloriani, and G. Vernazza, editors, Proceedings of the International Conference on Image Analysis and Processing (ICIAP), San Remo, volume 974 of LNCS, pages 68–73, Berlin, 1995. Springer.

[104] J. Suutala, S. Pirttikangas, J. Riekki, and J. Röning. Reject-optional LVQ-based two-level classifier to improve reliability in footstep identification. In A. Ferscher and F. Mattern, editors, Pervasive Computing, Proccedings on the Second International Conference PERVASIVE, Vienna, volume 3001 of LNCS, pages 182–187. Springer, 2004.

[105] G. Fumera and F. Roli. Support vector machines with embedded reject option. In S.-W. Lee and A. Verri, editors, Proceeedings of the First Interantional Workshop on Pattern Recognition with Support Vector Machines, Niagara Falls, volume 2388 of LNCS, pages 68–82. Springer, 2002.

[106] C.K. Chow. On optimum recognition error and reject tradeoff. IEEE Transactions in Information Theory, 16(1):41–46, 1970.

[107] C.K. Chow. An optimum character recognition system using decision functions. IRE Transactions on Electronic Computers, EC-6:247–254, 1957.

[108] T. Villmann, M. Kaden, D. Nebel, and M. Biehl. Learning vector quantization with adaptive cost-based outlier-rejection. In G. Azzopardi and N. Petkov, editors, Proceedings of 16th International Conference on Computer Analysis of Images and Pattern, CAIP 2015, Valetta - Malta, volume Part II of LNCS 9257, pages 772 – 782, Berlin-Heidelberg, 2015. Springer.

[109] T. Villmann, M. Kaden, A. Bohnsack, S. Saralajew, J.-M. Villmann, T. Drogies, and B. Hammer. Self-adjusting reject options in prototype based classification. In E. Merényi, M.J. Mendenhall, and P. O'Driscoll, editors, Advances in Self-Organizing Maps and Learning Vector Quantization: Proceedings of 11th International Workshop WSOM 2016, volume 428 of Advances in Intelligent Systems and Computing, pages 269–279, Berlin-Heidelberg, 2016. Springer.

[110] L. Fischer and T. Villmann. A probabilistic classifier model with adaptive rejection option. Machine Learning Reports, 10(MLR-01-2016):1–16, 2016. ISSN:1865-3960, http://www.techfak.uni-bielefeld.de/~fschleif/mlr/mlr_01_2016.pdf.

[111] V. Vovk, A. Gammerman, and G. Shafer. Algorithmic learning in a random world. Springer, Berlin, 2005.

[112] X. Zhu, F.-M. Schleif, and B. Hammer. Adaptive conformal semi-supervised vector quantization for dissimilarity data. Pattern Recognition Letters, 49:138–145, 2014.

[113] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, and P. Vincent. Why does unsupervised pretraining help deep learning. Journal of Machine Learning Research, 11:625–660, 2010.

[114] D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber. Multi-column deep neural network for traffic sign classification. Neural Networks, 32:333–338, 2012.

[115] Helge Ritter, Thomas Martinetz, and Klaus Schulten. Neural Computation and Self-Organizing Maps: An Introduction. Addison-Wesley, Reading, MA, 1992.

Prof. **Thomas Villmann** holds a diploma degree in Mathematics, received his Ph.D. in Computer Science in 1996 and his habilitation as well as venia legendi in the same subject in 2005, all from the University of Leipzig, Germany. From 1997 to 2009 he led the computational intelligence group of the Clinic for Psychotherapy at Leipzig University. Since 2009 he is a full Professor for Technomathematics/ Computational Intelligence at the University of Applied Sciences Mittweida, (Saxonia), Germany. He is founding member of the German chapter of European Neural Network Society (GNNS) and its president since 2011. Further he leads the Institute of Computational Intelligence and Intelligent Data Analysis e.V. in Mittweida, Germany and the Computational Intelligence Group at the University of Applied Sciences Mittweida. He acts as an associate editor for IEEE Transactions on Neural Networks and Learning Systems, Neural Processing Letters and for Computational Intelligence and Neuroscience. His research focus includes the theory of prototype based clustering and classification, non-standard metrics, information theoretic and similarity based learning, statistical data analysis and their application in pattern recognition, data mining and knowledge discovery for use in medicine, bioinformatics, remote sensing, hyperspectral analysis, forensics and others. Prof. Villmann was awarded several times for his research results. In 2016 he received the Wissenschaftspreis of the University of Applied Sciences Mittweida.

Andrea Bohnsack studied mathematics in Leipzig and Essen, Germany and received her diploma degree in 1997 from University Duisburg-Essen. Currently, she is a teacher for mathematics and computer science at an occupational highschool in Döbeln-Mittweida. Further, she is an associated member of the Computational Intelligence Group at the University of Applied Sciences in Mittweida. Her research interest are mathematical aspects of machine learning and classification.

Dr. **Marika Kaden** (formerly Kästner) holds a diploma in Applied Mathematics and M.Sc. degree in Discrete and Computer-oriented Mathematics, both from the University of Applied Sciences in Mittweida, Germany, in 2008 and 2010, respectively. In 2016 she received a PhD from the University Leipzig (Germany) in Computer Science. Since 2010 she is a member of the Computational Intelligence Group at the University of Applied Sciences Mittweida, currently as a postdoc researcher. Her research areas include optimized prototype-based classification learning, integration of expert knowledge, functional data processing and properties of dissimilarity structures.