University of Groningen

Learning vector quantization classifiers for ROC-optimization

Villmann, T.; Kaden, M.; Hermann, W.; Biehl, M.

Published in:
Computational Statistics

DOI:
[10.1007/s00180-016-0678-y](https://doi.org/10.1007/s00180-016-0678-y)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2018

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Villmann, T., Kaden, M., Hermann, W., & Biehl, M. (2018). Learning vector quantization classifiers for ROC-optimization. *Computational Statistics*, 33(3), 1173-1194. <https://doi.org/10.1007/s00180-016-0678-y>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Learning vector quantization classifiers for ROC-optimization

T. Villmann¹ · M. Kaden¹ · W. Hermann² ·
M. Biehl³

Received: 29 May 2015 / Accepted: 17 August 2016 / Published online: 27 August 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract This paper proposes a variant of the generalized learning vector quantizer (GLVQ) optimizing explicitly the area under the receiver operating characteristics (ROC) curve for binary classification problems instead of the classification accuracy, which is frequently not appropriate for classifier evaluation. This is particularly important in case of overlapping class distributions, when the user has to decide about the trade-off between high true-positive and good false-positive performance. The model keeps the idea of learning vector quantization based on prototypes by stochastic gradient descent learning. For this purpose, a GLVQ-based cost function is presented, which describes the area under the ROC-curve in terms of the sum of local discriminant functions. This cost function reflects the underlying rank statistics in ROC analysis being involved into the design of the prototype based discriminant function. The resulting learning scheme for the prototype vectors uses structured inputs, i.e. ordered pairs of data vectors of both classes.

Keywords Learning vector quantization · ROC analysis · AUC optimization

1 Introduction

Classification learning belongs to the most important tasks in machine learning. The respective mathematical basis is the Bayesian decision theory (Berger 1993). Bayes

✉ T. Villmann
thomas.villmann@hs-mittweida.de

¹ Computational Intelligence Group, University of Applied Sciences Mittweida, Mittweida, Germany

² Abt. Neurologie, Paracelsus-Klinikum Zwickau, Zwickau, Germany

³ Johann-Bernoulli-Institute for Mathematics and Computer Sciences, University Groningen, Groningen, The Netherlands

classifiers assume the knowledge or at least a good estimate of the class distributions. Yet, estimation of data densities in high-dimensional data may become difficult, imprecise and unstable (Biehl et al. 2011). Therefore, frequently the classification accuracy or approximations thereof are maximized instead of direct optimization of the Bayes criteria (Baldi et al. 2000; Bishop 2006; Duda and Hart 1973; Haykin 1994). Prominent machine learning examples are multilayer perceptrons, deep networks, decision trees, Bayesian classifiers, support vector machines (SVM) or learning vector quantizers (LVQ) (Bengio 2009; Schmidhuber 2015; LeCun et al. 2015; Huaichun et al. 1998; Kohonen 1986; Mitchell 1997; Quinlan 1993; Schölkopf and Smola 2002), whereas classical statistical discriminant analysis is mainly dominated by linear or quadratic discriminant analysis (LDA/QDA) (McLachlan 1992; Strickert et al. 2009; Yu et al. 1990) or the more general Fisher discriminant analysis (FDA), which requires weaker mathematical assumptions than LDA and QDA (Fisher 1936).

Among the machine learning approaches, prototype-based classifiers like SVM and LVQ allow an intuitive and easy interpretation of the classifier model (Cortes and Vapnik 1995; Cristianini and Shawe-Taylor 2000; Kohonen 1995; Hammer et al. 2014; Shawe-Taylor and Cristianini 2004; Vapnik 1998). Both models can be seen as classification margin maximizer (Hammer and Villmann 2002; Steinwart 2001). In SVM the model seeks for those data points in the training data, which determine the class borders. These data samples are denoted as support vectors in this model. Finally, the SVM classification problem can be reformulated in terms of the dual space such that it reduces to a convex optimization problem in that space (Shawe-Taylor and Cristianini 2004). Non-linear SVM for non-linear classification problems are obtained using the kernel approach (Schölkopf and Smola 2002). Alternatively, LVQ was originally designed to have class-typical prototype vector for the data approximating a Bayes decision. Yet, the resulting heuristic optimization strategies for the randomly initialized prototypes are usually compared in terms of accuracy (Kohonen 1995). The basic ingredients of these strategies are the data driven *attraction* to or the *repulsion* away from a given training data sample of the prototypes depending on their class coincidences. The heuristics differ in how and when one has to apply those attraction and repulsion operations for the prototypes. To overcome these heuristics, a gradient descent scheme was presented keeping the basic idea of attraction and repulsion in LVQ. The main goal in this approach is still to maximize the classification accuracy. More precisely, the cost function to be minimized in this generalized LVQ (GLVQ) is a soft approximation of the classification error (Sato and Yamada 1996). The exact classification error is obtained only in the limit (Kaden et al. 2015). GLVQ with more than one prototype per class can be seen as a non-linear classifier. However, also kernelized GLVQ versions are known keeping gradient descent learning strategy (Villmann et al. 2015).

Yet, the classification error may be misleading, in particular, if the data classes are imbalanced sampled. For those cases, other statistical quality measures for classification might be more appropriate. Examples for the binary case are the *true positive rate* (TPR) and the *false positive rate* (FPR) or the well-known F_β -measure, balancing the geometric and the arithmetic mean of precision and recall. For classifiers based on a continuous discriminant function receiver operating characteristics (ROC) analysis and evaluation of the related ROC curves are alternative tools (Bradley 1997; Huang

and Ling 2005). In particular, the area under the ROC-curve (AUC) provides a powerful measure for model comparisons also in the case of overlapping class distributions. Otherwise, optimized ROC-curves for those problems allow the user to decide the trade-off between high true-positive rate and good false-positive performance (Hanley and McNeil 1982; Lasko et al. 2005; Santos-Pereira and Pires 2005).

Several attempts were made to design machine learning classifiers, which *directly* optimize the AUC including gradient descent learning (Herschtal and Raskutti 2004), approximated AUC optimization (Calders and Jaroszewicz 2007), reject option optimization (Landgrebe et al. 2006; Villmann et al. 2015, 2016), AUC optimization by linear programming (Ataman et al. 2006) or ranking based optimization (Güvenir and Kurtcepe 2013), to name just a few of the recently proposed approaches. Unfortunately, these methods are neither comparable to SVM performance nor as intuitive as GLVQ. For *linear* SVM, a modification was proposed optimizing the AUC (Brefeld and Scheffer 2005; Rakotomamonjy 2004). To our best knowledge, non-linear versions are not proposed so far.

In this paper we propose a variant of GLVQ maximizing the AUC based on a gradient descent learning scheme. For this purpose, the probabilistic interpretation of the AUC is used and a respective cost function for GLVQ learning is derived preserving the main ideas of GLVQ which consist in prototype based model and gradient descent learning. Thus the intuitive understanding and interpretation of this approach is guaranteed. We denote this GLVQ variant as ROC-LVQ.

The remainder of the paper is as follows: First, we briefly review the ideas of learning vector quantization on the focusing on the particular aspects of GLVQ. Thereafter, we reconsider the ROC-theory emphasizing the probabilistic interpretation of the AUC. Both aspects are combined in the sequel resulting in the new ROC-LVQ approach. Numerical experiments demonstrate the behavior of the ROC-LVQ approach for synthetic and bio-medical examples.

2 Classification by learning vector quantization

Learning vector quantization (LVQ) models are prototype-based adaptive classifiers for processing vectorial data (Kohonen 1995). Training samples are assumed to be of the form $\mathbf{v} \in V \subseteq \mathbb{R}^n$ with class labels $x_{\mathbf{v}} = x(\mathbf{v}) \in \mathcal{C} = \{1, \dots, C\}$. The set of prototypes $W = \{\mathbf{w}_j \in \mathbb{R}^n, j = 1 \dots M\}$ contains representatives of the classes carrying prototype labels $y_j \in \mathcal{C}$. Classification decisions for unknown data samples $\tilde{\mathbf{v}}$ are usually made according to a winner take all rule, i.e.

$$x_{\tilde{\mathbf{v}}} := y_{s(\tilde{\mathbf{v}})} \text{ with } s(\tilde{\mathbf{v}}) = \operatorname{argmin}_j (d(\tilde{\mathbf{v}}, \mathbf{w}_j))$$

where $d(\tilde{\mathbf{v}}, \mathbf{w}_j)$ is a dissimilarity measure in the data space, frequently chosen as the Euclidean distance. LVQ training amounts to distributing the prototypes in the data space such that the classification error is reduced.

KOHONEN originally proposed several LVQ learning schemes for the Euclidean data space (Kohonen 1986, 1988). They all are based on attraction and repulsion of prototypes depending on their distance to the presented training sample and their class

coincidence (Kohonen 1990). Formally, the basic LVQ learning scheme consists of the attracting vector shift moving the winning prototype towards the presented input vector by

$$\Delta \mathbf{w}_{s(\mathbf{v})} = \varepsilon \cdot (\mathbf{v} - \mathbf{w}_{s(\mathbf{v})}) \text{ iff } x_{\mathbf{v}} = y_{s(\mathbf{v})} \quad (1)$$

for a correctly evaluated training sample $\mathbf{v}$ or a respective punishing (repulsion)

$$\Delta \mathbf{w}_{s(\mathbf{v})} = -\varepsilon \cdot (\mathbf{v} - \mathbf{w}_{s(\mathbf{v})}) \text{ iff } x_{\mathbf{v}} \neq y_{s(\mathbf{v})} \quad (2)$$

in case of a misclassification. The parameter $0 < \varepsilon \ll 1$ is denoted as learning rate and controls the strength of learning for a presented data vector $\mathbf{v}$.

The intuitive learning rules are obtained to achieve a good approximation of a Bayes decision based on vector quantization for class distribution approximation (Kohonen 1992).

Although LVQ algorithms frequently perform very well, the overall minimization of the classification error is not guaranteed, in particular, when the classes distributions overlap. Yet, all these original LVQ variants do not optimize any cost function. This disadvantage was overcome by SATO&YAMADA, who introduced a LVQ variant minimizing a soft version of the classification error while keeping the simple attraction and repulsing scheme of LVQ (Sato and Yamada 1996). This variant is called the *generalized* LVQ (GLVQ) and realizes a stochastic gradient descent learning on the objective function

$$E(W, f) = \frac{1}{2} \sum_{\mathbf{v} \in V} f(\mu(\mathbf{v})) \quad (3)$$

approximating the classification error (Sato and Yamada 1996). Here, the function

$$\mu(\mathbf{v}) = \frac{d^+(\mathbf{v}) - d^-(\mathbf{v})}{d^+(\mathbf{v}) + d^-(\mathbf{v})} \quad (4)$$

is the so-called classifier or discriminant function with $d^+(\mathbf{v}) = d(\mathbf{v}, \mathbf{w}^+)$ denoting the dissimilarity between the data vector $\mathbf{v}$ and the closest prototype $\mathbf{w}^+$ with the same class label $y_{s^+} = x_{\mathbf{v}}$, while $d^-(\mathbf{v}) = d(\mathbf{v}, \mathbf{w}^-)$ is the distance from the best matching prototype $\mathbf{w}^-$ with a class label y_{s^-} different from $x_{\mathbf{v}}$. The *modulation function* f in (3) is a monotonically increasing function usually chosen as a sigmoid or the identity function. A typical choice is the Fermi function

$$f_{\theta}(z) = \frac{1}{1 + a \cdot \exp\left(-\frac{(z-z_0)}{2\theta^2}\right)} \quad (5)$$

with $z_0 = 0$ and $a = 1$ as standard parameter values. The parameter θ determines the slope of f_{θ} but is frequently fixed as $\theta = 1$.

Stochastic gradient descent learning (SGDL) in GLVQ performs update steps of the form

$$\Delta \mathbf{w}^{\pm} = -\varepsilon(t) \cdot \frac{\partial f_{\theta}(\mu(\mathbf{v}))}{\partial \mu(\mathbf{v})} \cdot \frac{\partial \mu(\mathbf{v})}{\partial d^{\pm}(\mathbf{v})} \cdot \frac{\partial d^{\pm}(\mathbf{v})}{\partial \mathbf{w}^{\pm}} \quad (6)$$

for a randomly chosen data sample $\mathbf{v}$. Here, t denotes the time step of learning and $\varepsilon(t) > 0$ is the learning rate.¹ We remark that for the squared Euclidean distance we obtain

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

such that the GLVQ prototype update (6) becomes the scaled vector shifts (1) and (2) originally proposed by Kohonen with the local scaling factor $\xi^{\pm}(\mathbf{v}) = \frac{\partial f_{\theta}(\mu(\mathbf{v}))}{\partial \mu(\mathbf{v})} \cdot \frac{\partial \mu(\mathbf{v})}{\partial d^{\pm}(\mathbf{v})}$. However, more general differentiable dissimilarity measure can also be applied such as scaled Euclidean or kernel distances (Hammer and Villmann 2002; Villmann et al. 2015).

It turns out that GLVQ belongs to family of margin optimizer approaches in classification learning. In difference to SVM, which optimize the separation margin, GLVQ maximizes the hypothesis margin $M(\mathbf{v}, x(\mathbf{v})) = d^{-}(\mathbf{v}) - d^{+}(\mathbf{v})$ (Crammer et al. 2003; Hammer et al. 2005).

We remark that the classifier function $\mu(\mathbf{v}) \in [-1, 1]$ delivers a negative value for correct classification, i.e. $d^{+}(\mathbf{v}) < d^{-}(\mathbf{v})$. Further, the cost function (3) is a soft version of the classification error, which becomes crisp in the limit $\theta \searrow 0$ of the sigmoid f_{θ} (5). This corresponds to a transition of this sigmoid to the Heaviside function

$$H(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{else} \end{cases} \quad (7)$$

such that (3) really counts the misclassification in this setting. However, gradient descent learning is only preserved for non-vanishing values θ . For $0 < \theta \ll 1$ good approximations can be achieved, which realizes a *border sensitive classification learning* (BSCL) for GLVQ (BS-GLVQ) (Kästner et al. 2013). For this setting only those data samples $\mathbf{v}$ contribute significantly to the prototype learning, which are located nearby the class borders. For mathematical details we refer to (Kaden et al. 2015).

As we have seen just before, in the limit $\theta \searrow 0$ the cost function $E(W, f)$ from (3) becomes $E(W, H)$ counting exactly the *misclassifications*. Considering a two-class problem with a positive class A and a negative class B , these misclassifications can be further distinguished into the *false positives* and *false negatives* according to the contingency table Table 1.

¹ To ensure convergence of SGDL one has to require that $\sum_{t=1}^{\infty} \varepsilon(t) = \infty$, whereas $\sum_{t=1}^{\infty} \varepsilon^2(t) < \infty$ (Robbins and Monro 1951; Graf and Lushgy 2000). In practice, frequently the learning rate is set to be constant to a small positive value delivering also convergent behavior without loss of quality, i.e. $\varepsilon(t) = \varepsilon \ll 1$ (Haykin 1994). If it is not declared otherwise we take this latter option.

Table 1 Contingency table/confusion matrix

Labels	True			
		<i>A</i>	<i>B</i>	
Predicted	<i>A</i>	<i>TP</i>	<i>FP</i>	$\hat{N}_+$
	<i>B</i>	<i>FN</i>	<i>TN</i>	$\hat{N}_-$
		N_+	N_-	N

TP True positives, *FP* false positives, *TN* true negatives, *FN* false negatives, $N_{\pm}$ number of true positive/negative data, $\hat{N}_{\pm}$ number of predicted positive/negative samples

Yet, simple counting of general misclassifications is not always an appropriate evaluation criterion for classifier models, in particular, if the data are imbalanced (Sachs 1992). In statistical analysis contingency table evaluations are well-known to deal with this problem more properly. Several measures were developed to judge the classification quality based on the confusion matrix emphasizing different aspects. For example, *precision* π and *recall* ρ , defined as

$$\pi = \frac{TP}{TP + FP} = \frac{TP}{\hat{N}_+} \quad (8)$$

and

$$\rho = \frac{TP}{TP + FN} = \frac{TP}{N_+} \quad (9)$$

respectively, are used in the widely applied F_{β} -measure

$$F_{\beta} = \frac{(1 + \beta^2) \cdot \pi \cdot \rho}{\beta^2 \cdot \pi + \rho} \quad (10)$$

developed by C.J. VAN RIJSBERGEN (Rijsbergen 1979). Here, *TP* is the number of true positives, *FP* counts the false positives whereas *TN* and *FN* refer to the number of true and false negatives, respectively. $\hat{N}_{\pm}$ denotes the number of predicted (by the classifier) positive/negative samples. The recall ρ is also denoted as *true positive rate* (*TPR*) whereas the quantity

$$FPR = \frac{FP}{N_-} \quad (11)$$

is called *false positive rate*.

Recently, GLVQ variants were proposed, which explicitly optimize those statistical quality measures derived from the confusion matrix (Kaden et al. 2014). These measures might be preferred if the classes are imbalanced or if optimization of these measures is explicitly required like in medical application problems, where sensitivity and specificity are frequently favored over classification accuracy. Another criterion

Fig. 1 Visualization of the ROC space with classifier performances. *Left* 5 classifiers are displayed according to their performances. *A* nearly perfect classifier, *B* ‘conservative’ classifier, *C* ‘liberal’ classifier, *D* random guess classifier, *E* worse than random guess classifier (adapted from [Fawcett 2006](#)); *right* ROC curve for a classifier with continuous discriminant function and threshold parameter γ . Different γ -values correspond to different classifier performances generating the ROC-curve. The area under the ROC-curve (AUC) is equal to the probability P_{AB} that a classifier will rank a randomly chosen *A*-instance higher than a randomly chosen *B*-instance

of classifier performance and comparison frequently applied in bio-medical problems is the receiver operating characteristics ([Hanley and McNeil 1982, 1983](#); [Keilwagen et al. 2014](#); [Lasko et al. 2005](#)), which will be studied in relation to GLVQ learning in the next section.

3 Receiver operation characteristic curve optimization and GLVQ

3.1 Receiver operation characteristic—basic principles

The Receiver operation characteristic (ROC) is an important tool for performance comparison of binary classifiers. In particular, ROC graphs are useful tools for visual comparison of classifier performances. The two-dimensional ROC space is spanned along the axes TPR and FPR . In [Fig. 1](#)(left), a ROC graph with five classifiers *A* through *E* is depicted.

Roughly speaking, the more north-west a classifier is positioned in the ROC graph the better is the performance. According to [Fawcett \(2006\)](#), classifiers located on the left hand-side of the ROC graph may be seen as conservative avoiding false positives whereas classifiers appearing on the right hand-side may be thought of as ‘liberal’ preferring a high TPR -value while accepting high FPR -values. The diagonal represents the random guess. Classifiers depicted below this line perform worse than random guess. For detailed interpretations we refer to [Fawcett \(2006\)](#).

Binary classifiers with a continuous or rank-based discriminant function ϑ usually generate the classification decision for a given sample $\mathbf{v}$ in dependence on a threshold parameter γ . If $\vartheta(\mathbf{v}) > \gamma$ is valid, the first class *A* is selected. Otherwise, the sample is assigned to the opposite class *B*. Each threshold value corresponds to a certain classifier producing a different point in the ROC space preferring either class *A* for

low γ -values or B . All points form the ROC-curve, see Fig. 1(right). For the GLVQ discriminant function $\mu(\mathbf{v})$ in (4) one has to consider

$$\mu_\gamma(\mathbf{v}) = \frac{d^+(\mathbf{v}) - d^-(\mathbf{v})}{d^+(\mathbf{v}) + d^-(\mathbf{v})} - \gamma \quad (12)$$

with $\gamma \in [-1, 1]$ as a respective instantiation.

3.2 AUC for classifiers with continuous output in machine learning

ROC-analysis is a standard tool in machine learning (ML), if models with a continuous valued discrimination function are considered (Bradley 1997; Huang and Ling 2005). An important quantity to compare those classifiers is the *area under the ROC-curve* (AUC). A classifier is considered superior if it delivers a higher value of the AUC. The AUC has a probabilistic interpretation: Suppose the binary classification problem for classes A and B and related data sets V_A and V_B with cardinalities $\#V_A$, $\#V_B$, respectively. Then the AUC is equal to the probability P_{AB} that the classifier will rank a randomly chosen A -instance $\mathbf{v}_A \in V_A$ higher than a randomly chosen B -instance $\mathbf{v}_B \in V_B$ (Hanley and McNeil 1982; Wilcoxon 1945). Thus, following (Brefeld and Scheffer 2005), the AUC refers to the true distribution of positive and negative instances. Hence, it can be estimated using sampling. The normalized Wilcoxon–Mann–Whitney statistics (Wilcoxon 1945; Mann and Whitney 1947) reveals the maximum likelihood of the true AUC for a given classifier (Yan et al. 2003). More precisely, the probability P_{AB} can be estimated using sample subsets V_A and V_B . In particular, the (local) ordering function

$$O(\mathbf{v}_{AB}) = H(\vartheta(\mathbf{v}_A) - \vartheta(\mathbf{v}_B)) \quad (13)$$

is considered depending on an *ordered pair*

$$\mathbf{v}_{AB} = (\mathbf{v}_A, \mathbf{v}_B) \quad (14)$$

of samples and ϑ is a general real-valued discriminant function (Biehl et al. 2014). H is the Heaviside function from (7). We remark that $\mathbf{v}_{AB}$ is an element of the Cartesian data space $V_{AB} = V_A \times V_B$. By means of this ordering function we can approximate P_{AB} by

$$\hat{P}_{AB} = \frac{1}{\#V_{AB}} \sum_{\mathbf{v}_{AB} \in V_{AB}} O(\mathbf{v}_{AB}) \quad (15)$$

converging to the true value P_{AB} in the limit $\#V_{AB} \rightarrow \infty$ according to the underlying rank statistics (Mann and Whitney 1947; Wilcoxon 1945).

3.3 A cost function for AUC maximization based on GLVQ

As pointed out in the introduction, several ML approaches were developed to maximize AUC directly. Yet, for prototype based classification based on GLVQ, which can be seen as a robust variant of the nearest-neighbor classifier (Kaden et al. 2014), a direct optimization scheme for AUC is not known so far. As we will show in this chapter, the GLVQ can be easily adapted for AUC optimization using the probabilistic interpretation. The resulting algorithm is denoted as ROC-LVQ.

For this purpose, the general discriminant function ϑ in (13) is replaced by a discriminant function μ_{AB} specifically designed in the context of the GLVQ and, hence, based on the prototypes used in GLVQ. In particular, we define for a given data vector $\mathbf{v}$ the prototype based discriminant function

$$\mu_{AB}(\mathbf{v}, \gamma) = \frac{d^B(\mathbf{v}) - d^A(\mathbf{v})}{d^A(\mathbf{v}) + d^B(\mathbf{v})} - \gamma \quad (16)$$

with $d^A(\mathbf{v}) = d^A(\mathbf{v}, \mathbf{w}_A^*(\mathbf{v}))$, and $\mathbf{w}_A^*(\mathbf{v})$ is the closest prototype to $\mathbf{v}$ responsible for class A . Analogously, $\mathbf{w}_B^*$ and $d^B(\mathbf{v})$ are defined in the same manner. The parameter γ plays the role of the threshold of this discriminant function preferring either class A or B as demanded for AUC analysis. Further, we replace the Heaviside function in the local ordering function $O(\mathbf{v}_{AB})$ from (13) by sigmoid function f_θ from (5) and obtain

$$O_\theta^{\mu_{AB}}(\mathbf{v}_{AB}, W) = f_\theta(\mu_{AB}(\mathbf{v}_A, \gamma) - \mu_{AB}(\mathbf{v}_B, \gamma)) \quad (17)$$

as the the soft approximation $O(\mathbf{v}_{AB})$ with GLVQ-inspired discriminant function depending on the prototypes W . Now we define

$$E_{ROC-LVQ}(\theta, V_{AB}, W) = -\frac{1}{\#V_{AB}} \sum_{\mathbf{v}_{AB} \in V_{AB}} O_\theta^{\mu_{AB}}(\mathbf{v}_{AB}, W) \quad (18)$$

as cost function to be minimized parametrized by the slope parameter θ of the sigmoid function f_θ from (5). Obviously, forcing $\theta \searrow 0$ in (13) leads to the limit

$$E_{ROC-LVQ}(\theta, V_{AB}, W) \xrightarrow{\theta \searrow 0} P_{AB}, \quad (19)$$

which can be again interpreted as border sensitive learning as described in Sect. (2) for GLVQ. Further, using the derivatives

$$\frac{\partial \mu_{AB}(\mathbf{v}, \gamma)}{\partial \mathbf{w}_A^*(\mathbf{v})} = \frac{2 \cdot d^B(\mathbf{v})}{(d^A(\mathbf{v}) + d^B(\mathbf{v}))^2} \cdot \frac{\partial d^A(\mathbf{v})}{\partial \mathbf{w}_A^*(\mathbf{v})} \quad (20)$$

and

$$\frac{\partial \mu_{AB}(\mathbf{v}, \gamma)}{\partial \mathbf{w}_B^*(\mathbf{v})} = -\frac{2 \cdot d^A(\mathbf{v})}{(d^A(\mathbf{v}) + d^B(\mathbf{v}))^2} \cdot \frac{\partial d^B(\mathbf{v})}{\partial \mathbf{w}_B^*(\mathbf{v})} \quad (21)$$

we can calculate the gradients of the GLVQ-adapted ordering function regarding to both $\mathbf{v}_A$ and $\mathbf{v}_B$, respectively:

$$\frac{\partial O_{\theta}^{\mu_{AB}}(\mathbf{v}_{AB}, W)}{\partial \mathbf{w}_A^*(\mathbf{v}_A)} = \frac{\partial f_{\theta}}{\partial z} \Big|_z \cdot \left(\frac{\partial \mu_{AB}(\mathbf{v}_A, \gamma)}{\partial \mathbf{w}_A^*(\mathbf{v}_A)} - \frac{\partial \mu_{AB}(\mathbf{v}_B, \gamma)}{\partial \mathbf{w}_A^*(\mathbf{v}_A)} \right) \quad (22)$$

$$\frac{\partial O_{\theta}^{\mu_{AB}}(\mathbf{v}_{AB}, W)}{\partial \mathbf{w}_A^*(\mathbf{v}_B)} = \frac{\partial f_{\theta}}{\partial z} \Big|_z \cdot \left(\frac{\partial \mu_{AB}(\mathbf{v}_A, \gamma)}{\partial \mathbf{w}_A^*(\mathbf{v}_B)} - \frac{\partial \mu_{AB}(\mathbf{v}_B, \gamma)}{\partial \mathbf{w}_A^*(\mathbf{v}_B)} \right) \quad (23)$$

$$\frac{\partial O_{\theta}^{\mu_{AB}}(\mathbf{v}_{AB}, W)}{\partial \mathbf{w}_B^*(\mathbf{v}_A)} = \frac{\partial f_{\theta}}{\partial z} \Big|_z \cdot \left(\frac{\partial \mu_{AB}(\mathbf{v}_A, \gamma)}{\partial \mathbf{w}_B^*(\mathbf{v}_A)} - \frac{\partial \mu_{AB}(\mathbf{v}_B, \gamma)}{\partial \mathbf{w}_B^*(\mathbf{v}_A)} \right) \quad (24)$$

$$\frac{\partial O_{\theta}^{\mu_{AB}}(\mathbf{v}_{AB}, W)}{\partial \mathbf{w}_B^*(\mathbf{v}_B)} = \frac{\partial f_{\theta}}{\partial z} \Big|_z \cdot \left(\frac{\partial \mu_{AB}(\mathbf{v}_A, \gamma)}{\partial \mathbf{w}_B^*(\mathbf{v}_B)} - \frac{\partial \mu_{AB}(\mathbf{v}_B, \gamma)}{\partial \mathbf{w}_B^*(\mathbf{v}_B)} \right) \quad (25)$$

with $z = \mu_{AB}(\mathbf{v}_A, \gamma) - \mu_{AB}(\mathbf{v}_B, \gamma)$.

In consequence, GLVQ-like stochastic gradient optimization is possible also for the ROC cost function $E_{ROC-GLVQ}$ from (18) requiring the *structured input* $\mathbf{v}_{AB} = (\mathbf{v}_A, \mathbf{v}_B)$ during learning, which is already introduced in (14). However, learning by structured inputs is a new aspect in GLVQ learning causing a more complex update scheme. In difference to standard GLVQ, the stochastic gradient descent learning for structured inputs has to take place with respect to all four quantities $\mathbf{w}_A^*(\mathbf{v}_A)$, $\mathbf{w}_A^*(\mathbf{v}_B)$, $\mathbf{w}_B^*(\mathbf{v}_A)$, and $\mathbf{w}_B^*(\mathbf{v}_B)$ using the derivatives (22)–(25) of the GLVQ-adapted ordering function $O_{\theta}^{\mu_{AB}}$ from (17). Summarizing, we obtain the general SGDL scheme

$$\Delta \mathbf{w}_X^*(\mathbf{v}_Y) = \varepsilon \cdot \frac{\partial O_{\theta}^{\mu_{AB}}(\mathbf{v}_A, \mathbf{v}_B, W)}{\partial \mathbf{w}_X^*(\mathbf{v}_Y)} \quad (26)$$

according to

$$\frac{\partial_S E_{ROC-LVQ}(\theta, \mathbf{v}_{AB}, W)}{\partial \mathbf{w}_X^*(\mathbf{v}_Y)} = \frac{\partial O_{\theta}^{\mu_{AB}}(\mathbf{v}_A, \mathbf{v}_B, W)}{\partial \mathbf{w}_X^*(\mathbf{v}_Y)}$$

with the specifications A and B for X and Y accordingly for updating, as explained above. The resulting algorithm² is summarized in Algorithm 1.

² The algorithm is implemented in MATLABTM as the ROCGMLVQ-package (Vers. 1.7). It is available from the authors by personal request or via the webpage <https://www.cb.hs-mittweida.de/webs/villmann/research/tools-data.html>.

1: **procedure** ROC-LVQ(V , numIterations)

2: Initialize learning rate $0 < \varepsilon \ll 1$

3: Initialize prototype set $W = \{\mathbf{w}_1, \dots, \mathbf{w}_M\}$ randomly
(at least one prototype per class)

```

4:   for i = 1 : numIterations do

```

5: Select two data points $\mathbf{v}_A, \mathbf{v}_B$ with $x_{\mathbf{v}_A} = A$, $x_{\mathbf{v}_B} = B$ randomly

6: Determine prototypes $\mathbf{w}_A^*(\mathbf{v}_A)$, $\mathbf{w}_A^*(\mathbf{v}_B)$, $\mathbf{w}_B^*(\mathbf{v}_A)$, $\mathbf{w}_B^*(\mathbf{v}_B)$:

$$\mathbf{w}_A^*(\mathbf{v}_A) = \underset{\mathbf{w}_j \in W^A}{\operatorname{argmin}} d(\mathbf{v}_A, \mathbf{w}_j) \quad \text{with} \quad W^A = \{\mathbf{w}_j \in W | y(\mathbf{w}_j) = A\}$$

$$\mathbf{w}_A^*(\mathbf{v}_B) = \underset{\mathbf{w}_j \in W^A}{\operatorname{argmin}} d(\mathbf{v}_B, \mathbf{w}_j) \quad \text{with} \quad W^A = \{\mathbf{w}_j \in W | y(\mathbf{w}_j) = A\}$$

$$\mathbf{w}_B^*(\mathbf{v}_A) = \operatorname{argmin}_{\mathbf{w}_j \in W^B} d(\mathbf{v}_A, \mathbf{w}_j) \quad \text{with} \quad W^B = \{\mathbf{w}_j \in W | y(\mathbf{w}_j) = B\}$$

$$\mathbf{w}_B^*(\mathbf{v}_B) = \operatorname{argmin}_{\mathbf{w}_j \in W^B} d(\mathbf{v}_B, \mathbf{w}_j) \quad \text{with} \quad W^B = \{\mathbf{w}_j \in W | y(\mathbf{w}_j) = B\}$$

7: Calculate the quantities

$$d^A(\mathbf{v}_A) = d(\mathbf{v}_A, \mathbf{w}_A^*) \quad , \quad d^B(\mathbf{v}_A) = d(\mathbf{v}_A, \mathbf{w}_B^*)$$

and

$$d^A(\mathbf{v}_B) = d(\mathbf{v}_B, \mathbf{w}_A^*) \quad , \quad d^B(\mathbf{v}_B) = d(\mathbf{v}_B, \mathbf{w}_B^*)$$

8: Update prototypes $\mathbf{w}_A^*$ and $\mathbf{w}_B^*$ (see (26)):

$$\begin{aligned}\mathbf{w}_{A/B}^*(\mathbf{v}_A) &= \mathbf{w}_{A/B}^*(\mathbf{v}_A) + \varepsilon \cdot \Delta \mathbf{w}_{A/B}^*(\mathbf{v}_A) \\ \Delta \mathbf{w}_{A/B}^*(\mathbf{v}_A) &= \frac{\partial O_{\theta}^{\mu_{AB}}(\mathbf{v}_A, \mathbf{v}_B, W)}{\partial \mathbf{w}_{A/B}^*(\mathbf{v}_A)} \\ \mathbf{w}_{A/B}^*(\mathbf{v}_B) &= \mathbf{w}_{A/B}^*(\mathbf{v}_B) + \varepsilon \cdot \Delta \mathbf{w}_{A/B}^*(\mathbf{v}_B) \\ \Delta \mathbf{w}_{A/B}^*(\mathbf{v}_B) &= \frac{\partial O_{\theta}^{\mu_{AB}}(\mathbf{v}_A, \mathbf{v}_B, W)}{\partial \mathbf{w}_{A/B}^*(\mathbf{v}_B)}\end{aligned}$$

9: Decrease learning rate ε if this option is chosen according to footnote 1.

```

10:   end for

```

11: Return prototype set W

12: end procedure

 Springer

kernel distances, correlation measures or parametrized measures allowing relevance learning (Schneider et al. 2009; Strickert et al. 2008; Villmann et al. 2015).

4 Experiments

In this section we provide the application results of the presented approach for two illustrative artificial data sets AD1 and AD2 as well as three real world data sets obtained from the UCI database and in medical application domain.

4.1 Illustrating artificial data sets

The first data set used for demonstration is an artificial toy data set (AD1) of two-dimensional vectors with two slightly overlapping classes of different variability. In particular, one of the classes displays a small cluster of data separated from the two over-lapping clusters which represent the bulk of the data, see Fig. 2. The green class ‘o’ contains 526 samples whereas the magenta ‘◇’ class consists of 274 data.

We trained both GLVQ and ROC-LVQ each with two prototypes per class to give the models the chance to cover both classes adequately. A higher number of prototypes

Fig. 2 Visualization of the data set AD1 and simulation results. Distribution of the two classes (*green open circle* and *magenta open diamond*) together with the respective prototypes obtained from best GLVQ (*red plus* and *times*) and best ROC-LVQ (*blue open triangle* and *open downtriangle*) (color figure online)

Table 2 AUC-values and classification accuracies for the artificial data sets AD1 and AD2

	AD1		AD2	
	GLVQ	ROC-LVQ	GLVQ	ROC-LVQ
Accuracy in %	99.4	99.3	97.4	96.3
AUC	0.9942	0.9969	0.9922	0.9957

Fig. 3 ROC curves for the data set AD1 and best GLVQ (red dashed line) and ROC-LVQ (blue solid line) from Fig. 2, together with respective AUC-values—left normal plot, right—axes in logarithmic scale (color figure online)

would lead to vanishing effects of ROC-LVQ compared to GLVQ, because the expected number of misclassified data would decrease to zero in this case.

The prototypes were randomly initialized to be a data sample of the respective class to avoid dead units. The constant learning rate was $\varepsilon = 0.01$ and the models were trained within 3000 learning epochs, i.e. each training data vector was presented once in a random sequence per epoch. The results are shown in Table 2. We detect an improvement of the AUC value in case of ROC-LVQ compared to GLVQ while the accuracy, measured taking $\gamma = 0$ in (12) during the recall, is lowered. The related ROC-curves are depicted in Fig. 3.

Taking a closer look to the respective distributions of the prototypes we observe their different behaviors, 2. Whereas GLVQ tries to represent the outlier subset of the green class to ensure a high accuracy, ROC-LVQ better represents the bulk of this class and is more sensitive for the overlap region.

The second artificial data (AD2) set is also a two-dimensional data set of overlapping classes, however, consisting of non-linear data distributions with unbalanced number of class samples, see Fig. 4. Here, the green class ‘o’ comprises 200 samples whereas the magenta ‘◇’ class contains 606 data. Accordingly, we trained the models with 3 and 6 prototypes per class reflecting the class different class distributions. At least three prototypes per class are required to reflect the non-linearity in data, see Fig. 4. As before, a higher number of prototypes would lead to neglecting effects of the proposed algorithm compared to original GLVQ due to the extremely small number of misclassification to be expected in those settings.

Fig. 4 Visualization of the data set AD2 and simulation results. Distribution of the two classes (*green open circle* and *magenta open diamond*) together with the respective prototypes obtained from best GLVQ (*red plus* and *times*) and best ROC-LVQ (*blue open triangle* and *open downtriangle*) (color figure online)

Fig. 5 ROC curves for the data set AD2 and best GLVQ (*red dashed line*) and ROC-LVQ (*blue solid line*) from Fig. 2, together with respective AUC-values—*left* normal plot, *right*—axes in logarithmic scale (color figure online)

The remaining training parameter were set as for AD1. Also for this strong non-linear case, ROC-LVQ found a slightly better prototype distribution in terms of AUC, see Table 2. The better accuracy is achieved by GLVQ, however, accompanied by a lower AUC-value. The respective ROC-curves are depicted also in Fig. 5.

Thus ROC-LVQ did the expected job of AUC-optimization for both toy examples passing on the cost to the accuracy for the $\gamma = 0$ recall regime. For real world applications, the user has to decide, which γ -value fits best the expectations and requirements.

4.2 Real world application

In this subsection we present the application of the ROC-LVQ for real world data sets. Two of them are from the UCI dataset such that the reader can compare with other approaches. The third one is a classification problem in the medical domain regarding Wilson's disease (WD).

4.2.1 UCI data

We compare in this consideration the ROC-LVQ with GLVQ for two benchmark UCI-data sets: the Wisconsin-Breast-Cancer-data (WDBC) and the Indian diabetes data set (PIMA) (Blake and Merz 1998). The data sets contain 562 and 768 data vectors with 32 and 8 data dimensions, respectively, and each divided into two classes (healthy/ill). A detailed description can be found in Blake and Merz (1998).

The presented results are obtained from a five-fold cross validation repeated ten times for statistical evaluation. For each simulation we used only one prototype per class to keep the model complexity as low as possible. The learning rate was chosen to be constant $\varepsilon = 0.01$ for all runs and the overall learning time was set to be 1000 epochs. The results are depicted in Table 3.

We observe for both data sets increased AUC-values for ROC-LVQ compared with those from GLVQ. Although the improvements are only slightly, the increase is statistically significant. The test accuracy, however, remained at the same level within the statistical confidence.

For both examples PIMA and WDBC, the ROC-curves of the best GLVQ-result are compared with the best ROC-results (each for the first fold) in Figs. 6 and 7, respectively.

It is well-known that in general the classification accuracy of LVQ algorithms increases with increasing number of prototypes. This behavior should be expected also

Table 3 Accuracies and AUC-values for the test data sets of PIMA, WDBC and WD-problem together with the standard deviation obtained from 5-fold cross validation and 10 repetitions each

	PIMA		WDBC		WD	
	GLVQ	ROC-LVQ	GLVQ	ROC-LVQ	GLVQ	ROC-LVQ
Accuracy	74.9 %	74.2 %	93.2 %	93.9 %	86.6 %	77.1 %
	± 1.6 %	± 4.6 %	± 1.1 %	± 1.0 %	± 1.6 %	± 5.6 %
AUC	0.810	0.828	0.982	0.986	0.901	0.957
	± 0.023	± 0.030	± 0.010	± 0.008	± 0.035	± 0.007

Fig. 6 ROC-curves obtained from the best GLVQ-model in terms of the GLVQ cost function (3) (red dashed line) and the best ROC-LVQ in terms of the AUC-value (blue solid line) for the PIMA dataset, see text (color figure online)

Fig. 7 ROC-curves obtained from the best GLVQ-model in terms of the GLVQ cost function (3) (red dashed line) and the best ROC-LVQ in terms of the AUC-value (blue solid line) for the WBDC dataset, see text (color figure online)

for ROC-LVQ. Otherwise, the number of prototypes should be kept as low as possible to avoid overfitting. To investigate the influence of the number of prototypes regarding the performance of ROC-LVQ, we exemplarily compared ROC-LVQ and GLVQ for the PIMA dataset when increasing the number of prototypes per class. The results are given in Table 4. As expected, we perceive an improving accuracy for an increasing number of prototypes, which are almost equal for both GLVQ and ROC-LVQ. For the AUC-values, their differences between GLVQ and ROC-LVQ are decreased with increasing number of prototypes. We can conclude from this observation that for sufficient large number of prototypes ROC-LVQ and GLVQ behave similar.

4.2.2 Classification of Wilson's disease patients

This real world problem deals with a medical classification problem concerning detection of Wilson's disease (WD). WD is a autosomal-recessive disorder of cop-

Table 4 Accuracies and AUC-values for the test data sets of PIMA together with the standard deviation obtained from 5-fold cross validation with different number of prototypes (no repetitions compared to Table 3)

Prototypes	GLVQ			ROC-LVQ		
	1 1	2 2	5 5	1 1	2 2	5 5
Accuracy	73.1 %	78.3 %	82.3 %	73.1 %	78.3 %	82.7 %
	± 3.9 %	± 3.9 %	± 3.5 %	± 4.3 %	± 3.9 %	± 2.9 %
AUC	0.808	0.836	0.859	0.813	0.846	0.861
	± 0.016	± 0.030	± 0.025	± 0.019	± 0.029	± 0.031

Fig. 8 Examples of PET images for glucose consumption: from *left to right*—normal, non-neurological WD, neurological WD sample

per metabolism in the liver. In consequence, copper is accumulated in several brain regions causing neurophysiological impairments, see Fig. 8.

The disturbances lead to neurophysiological and fine-motoric impairments (Hermann et al. 2002, 2003). Beside other symptoms, one indication of disturbed neurophysiological abilities is a reduced glucose metabolism (Hermann et al. 2002). Generally, one has to distinguish between two phases in WD. The first *non-neurological* phase with weak clinical symptoms and unclear neurological disturbances is followed by the *neurological* phase with severe impairments in neurophysiology as well as fine-motoric abilities (Hermann et al. 2003). Hence, the weak neurological impairments in the non-neurological phase are difficult to detect. One way to detect impairments is determination of the glucose consumption in several brain areas based on a $^{18}\text{Fluorodesoxyglucose-Positron-Emission-Tomography}$ ($^{18}\text{FDG-PET}$) analysis. For the WD-investigation considered here, the glucose consumption in 11 human brain regions (*cerebellum, frontal lobe, parietal lobe, temporal lobe, occipital lobe, ant. cingulum, caudatus head, putamen, thalamus, midbrain, post. cingulum*) was assessed from $^{18}\text{FDG-PET}$ images and collected in a respective FDG-PET vector for analysis roughly as the averaged consumption in each brain region, the data pre-processing is described (Hermann et al. 2002) and citations therein.

Thus, the data set to be analyzed in this investigation contains 11-dimensional data vectors from 65 WD-patients, 18 of them are in the non-neurological phase (NN) and 47 are neurological (N). Additionally, 15 data vectors are from healthy

Table 5 Accuracies and AUC-values for the test data sets of the WD-problem together with the standard deviation obtained from 5-fold cross validation and 10 repetitions each

	GLVQ	ROC-LVQ
Accuracy	86.6 $\pm$ 1.6 %	77.1 $\pm$ 5.6 %
AUC	0.901 $\pm$ 0.035	0.957 $\pm$ 0.007

Fig. 9 ROC-curves obtained from GLVQ model with highest accuracy of all runs (*red dashed line*) and ROC-LVQ with highest AUC-value of all runs (*blue solid line*) for the WD data set, see text (color figure online)

volunteers. It is an extended data set already studied in [Hermann et al. \(2002\)](#). We trained a standard GLVQ as well as a ROC-LVQ, both using the Euclidean distance, for the binary classification problem to separate patient vectors from those of healthy volunteers with 2 prototypes for the healthy class and 4 for the patient class. This choice was done to reflect roughly the ratio of training data in the classes and to have at least two prototypes per class for more flexible adaptation compared to taking only one prototype for the minor class, which could be simply chosen as the class mean. However, adding more prototypes increases the probability of overfitting due to the low number of available training data.

The results presented in Table 5 are obtained from a 5-fold cross validation scheme with 10 repetitions in each fold starting with randomly initialized prototypes. The learning rate was constant $\varepsilon = 0.01$ and the learning time was 3000 epochs.

We observe an increasing AUC-value for ROC-LVQ in comparison to GLVQ, as expected. Otherwise, the averaged accuracy of ROC-LVQ is decreased with growing variance in comparison to GLVQ.

In Fig. 9, the ROC-curves for the best GLVQ model (in terms of accuracy) and the best ROC-LVQ (in terms of AUC) model of the first fold are depicted. The best GLVQ achieved an accuracy of $acc_{GLVQ} = 88.75\%$ with a respective AUC-value $AUC_{GLVQ} = 0.9200$. The best ROC-LVQ model yielded $auc_{ROC-LVQ} = 0.9682$ with an accompanied accuracy value $acc_{ROC-LVQ} = 83.75\%$. As expected from the medical knowledge, the classification errors mainly are caused by false classification of non-neurological patient vectors. Thus, the ROC-LVQ model provides the medical

expert the best recommendation model in dependence of the trade-off between true-positive and false-positive rate controlled by the γ -threshold.

5 Conclusion and remarks

We present in this article the mathematical framework for learning of prototype-based LVQ-classifiers to optimize the receiver operating characteristic. In particular, the AUC-value is maximized. For this purpose, the probabilistic interpretation of the area under the ROC-curve was used together with the underlying ordering function of the rank statistics. The resulting ROC-LVQ cost function reflects this ordering function in terms of the prototype-based discriminant function, which can be minimized using a stochastic gradient descent learning scheme. This translates to the idea of structured inputs for a GLVQ-like learning of the prototypes. We explained the mathematical framework for the algorithm. Illustrative toy applications as well as three real world classification tasks demonstrate the behavior of the ROC-LVQ algorithm in work.

Obviously, the ROC-LVQ can be easily combined with other advanced GLVQ-techniques like relevance and matrix learning for task dependent dissimilarity adaptation or kernelized variants (Hammer and Villmann 2002; Schneider et al. 2009; Villmann et al. 2014). Recently also a median variant of ROC-LVQ was proposed restricting the prototypes to be data points and applying a generalized EM-algorithm for prototype optimization (Nebel and Villmann 2015).

References

- Ataman K, Street WN, Zhang Y (2006) Learning to rank by maximizing AUC with linear programming. In: Proceedings of the IEEE international joint conference on neural networks (IJCNN). IEEE Press, pp 123–129
- Baldi P, Brunak S, Chauvin Y, Andersen C, Nielsen H (2000) Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics* 16(5):412–424
- Bengio Y (2009) Learning deep architectures for AI. *Found Trends Mach Learn* 2(1):1–127
- Berger JO (1993) Statistical decision theory and Bayesian analysis. Springer series in statistics, 3rd edn. Springer, New York
- Biehl M, Hammer B, Merényi E, Sperduti A, Villman T (2011) Learning in the context of very high dimensional data (Dagstuhl Seminar 11341). *Dagstuhl Rep* 1(8):67–95
- Biehl M, Kaden M, Stürmer P, Villmann T (2014) ROC-optimization and statistical quality measures in learning vector quantization classifiers. *Mach Learn Rep*, 8(MLR-01-2014):23–34, ISSN:1865-3960, http://www.techfak.uni-bielefeld.de/~fschleif/mlr/mlr_01_2014.pdf
- Bishop CM (2006) Pattern recognition and machine learning. Springer, New York
- Blake CL, Merz CJ (1998) UCI repository of machine learning databases. University of California, Dep. of Information and Computer Science, Irvine. <http://www.ics.edu/mllearn/MLRepository.html>
- Bradley AP (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recogn* 30(7):1149–1155
- Brefeld U, Scheffer T (2005) AUC maximizing support vector learning. In: Proceedings of ICML 2005 workshop on ROC analysis in machine learning, pp 377–384
- Calders T, Jaroszewicz S (2007) Efficient AUC optimization for classification. In: Kok JN, Koronacki J, de Mantaras R Lopez, Matwin S, Mladenic D, Skowron A (eds) Knowledge discovery in databases: PKDD 2007, volume 4702 of LNCS. Springer-Verlag, Berlin, pp 42–53
- Cortes C, Vapnik V (1995) Support vector network. *Mach Learn* 20:1–20

- Crammer K, Gilad-Bachrach R, Navot A, Tishby A (2003) Margin analysis of the LVQ algorithm. In: Becker S, Thrun S, Obermayer K (eds) *Advances in neural information processing (Proc. NIPS 2002)*, vol 15. MIT Press, Cambridge, pp 462–469
- Cristianini N, Shawe-Taylor J (2000) *An introduction to support vector machines and other Kernel-based learning methods*. Cambridge University Press, Cambridge
- Duda RO, Hart PE (1973) *Pattern Classification and scene analysis*. Wiley, New York
- Fawcett T (2006) An introduction to ROC analysis. *Pattern Recogn Lett* 27:861–874
- Fisher RA (1936) The use of multiple measurements in taxonomic problems. *Ann Eugen* 7(2):179–188
- Graf S, Lushgy H (2000) *Foundations of quantization for random vectors*. LNM-1730. Springer, Berlin
- Güvenir HA, Kurtcepe M (2013) Ranking instances by maximizing the area under ROC curve. *IEEE Trans Knowl Data Eng* 25(10):2356–2366
- Hammer B, Strickert M, Villmann T (2005) On the generalization ability of GRLVQ networks. *Neural Process Lett* 21(2):109–120
- Hammer B, Nebel D, Riedel M, Villmann T (2014) Generative versus discriminative prototype based classification. In: Villmann T, Schleif F-M, Kaden M, Lange M (eds) *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*. Springer, Berlin, pp 123–132
- Hammer B, Villmann T (2002) Generalized relevance learning vector quantization. *Neural Netw* 15(8–9):1059–1068
- Hanley JA, McNeil BJ (1982) The meaning and use of the area under a receiver operating characteristic. *Radiology* 143:29–36
- Hanley JA, McNeil BJ (1983) A method of comparing the area under receiver operating characteristic curves derived from the same case. *Radiology* 148(3):839–843
- Haykin Simon (1994) *Neural networks. A comprehensive foundation*. Macmillan, New York
- Hermann W, Barthel H, Hesse S, Villmann Th, Wagner A (2002) Korrelation der motorisch evozierten Potentiale mit dem striatalen Glukosestoffwechsel bei Patienten mit einem Morbus Wilson. *Aktuelle Neurol* 5:242–246
- Hermann W, Barthel H, Hesse S, Grahmann F, Kühn H-J, Wagner A, Villmann Th (2002) Comparison of clinical types of Wilson's disease and glucose metabolism in extrapyramidal motor brain regions. *J Neurol* 249(7):896–901
- Hermann W, Villmann Th, Grahmann F, Kühn HJ, Wagner A (2003) Investigation of fine motoric disturbances in Wilson's disease. *Neurol Sci* 23(6):279–285
- Herschtal A, Raskutti B (2004) Optimising area under the ROC curve using gradient descent. In: *Proceedings of the 21st international conference on machine learning*. Banff, pp 49–56
- Huaichun W, Dopazo J, Carazo JM (1998) Self-organizing tree growing network for classifying amino acids. *Bioinformatics* 14(4):376–377
- Huang J, Ling CX (2005) Using AUC and accuracy in evaluating learning algorithms. *IEEE Trans Knowl Data Eng* 17(3):299–310
- Kaden M, Hermann W, Villmann T (2014) Optimization of general statistical accuracy measures for classification based on learning vector quantization. In: Verleysen M (ed) *Proceedings of European symposium on artificial neural networks, computational intelligence and machine learning (ESANN'2014)*. Louvain-La-Neuve, Belgium, pp 47–52
- Kaden M, Lange M, Nebel D, Riedel M, Geweniger T, Villmann T (2014) Aspects in classification learning—review of recent developments in learning vector quantization. *Found Comput Decis Sci* 39(2):79–105
- Kaden M, Riedel M, Hermann W, Villmann T (2015) Border-sensitive learning in generalized learning vector quantization: an alternative to support vector machines. *Soft Comput* 19(9):2423–2434
- Kästner M, Riedel M, Strickert M, Hermann W, Villmann T (2013) Border-sensitive learning in kernelized learning vector quantization. In: Rojas I, Joya G, Cabestany J (eds) *Proceedings of the 12th international workshop on artificial neural networks (IWANN)*, volume 7902 of *LNCS*. Springer, Berlin, pp 357–366
- Keilwagen J, Grosse I, Grau J (2014) Area under precision-recall curves for weighted and unweighted data. *PLoS One* 9(3):1–13
- Kohonen T (1990) Improved versions of learning vector quantization. In: *Proceedings of IJCNN-90, international joint conference on neural networks*, vol I. Piscataway, IEEE Service Center, San Diego, pp 545–550
- Kohonen Teuvo (1986) *Learning vector quantization for pattern recognition*. Report TKK-F-A601, Helsinki University of Technology, Espoo

- Kohonen T (1988) Learning vector quantization. *Neural Netw* 1(Supplement 1):303
- Kohonen T (1992) Learning-vector quantization and the self-organizing map. In: Taylor JG, Mannion CLT (eds) *Theory and applications of neural networks*. Springer, London, pp 235–242
- Kohonen Teuvo (1995) *Self-organizing maps*, volume 30 of Springer series in information sciences. Springer, Berlin, Heidelberg (Second Extended Edition 1997)
- Landgrebe TCW, Tax D, Paclik P, Duin RPW (2006) The interaction between classification and reject performance for distance-based reject-option classifiers. *Pattern Recogn Lett* 27:908–917
- Lasko TA, Bhagwat JG, Zou KH, Ohno-Machado L (2005) The use of receiver operating characteristic curves in biomedical informatics. *J Biomed Inf* 38:404–415
- LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521:436–444
- Mann HB, Whitney DR (1947) On a test whether one of two random variables is stochastically larger than the other. *Ann Math Stat* 18:50–60
- McLachlan GJ (1992) *Discriminant analysis and statistical pattern recognition.*, Wiley series in probability and mathematical statistics: applied probability and statistics Wiley, New York
- Mitchell T (1997) *Machine learning*. mcgraw hill, New York
- Nebel D, Villmann T (2015) Median-LVQ for classification of dissimilarity data based on ROC-optimization. In: Verleysen M (ed) *Proceedings of the European symposium on artificial neural networks, computational intelligence and machine learning (ESANN'2015)*. Louvain-La-Neuve, Belgium, pp 1–6
- Quinlan JR (1993) *C4.5: programs for machine learning*. Morgan Kaufmann, San Francisco
- Rakotomamonjy A (2004) Optimizing area under ROC curve with SVMs. In: *Proceedings of the workshop on ROC analysis in artificial intelligence*, Hamburg, pp 71–80
- Rijsbergen CJ (1979) *Information retrieval*, 2nd edn. Butterworths, London
- Robbins H, Monro S (1951) A stochastic approximation method. *Ann Math Stat* 22:400–407
- Sachs L (1992) *Angewandte Statistik*, 7th edn. Springer Verlag, Berlin
- Santos-Pereira CM, Pires AM (2005) On optimal reject rules and ROC curves. *Pattern Recogn Lett* 26:943–952
- Sato A, Yamada K (1996) Generalized learning vector quantization. In: Touretzky DS, Mozer MC, Hasselmo ME (eds) *Advances in neural information processing systems 8*. Proceedings of the 1995 conference. MIT Press, Cambridge, pp 423–429
- Schmidhuber J (2015) Deep learning in neural networks: an overview. *Neural Netw* 61:85–117
- Schneider P, Hammer B, Biehl M (2009) Adaptive relevance matrices in learning vector quantization. *Neural Comput* 21:3532–3561
- Schölkopf B, Smola A (2002) *Learning with Kernels*. MIT Press, Cambridge
- Shawe-Taylor J, Cristianini N (2004) *Kernel methods for pattern analysis and discovery*. Cambridge University Press, Cambridge
- Steinwart I (2001) On the influence of the kernel on the consistency of support vector machines. *J Mach Learn Res* 2:67–93
- Strickert M, Schleif F-M, Seiffert U, Villmann T (2008) Derivatives of Pearson correlation for gradient-based analysis of biomedical data. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial* 37:37–44
- Strickert M, Keilwagenan J, Schleif F-M, Villmann T, Biehl M (2009) Matrix metric adaptation linear discriminant analysis of biomedical data. In: Cabestany J et al (eds) *Proceedings international workshop on artificial neural networks (IWANN) 2009*, volume 5517 of LNCS. Springer, Heidelberg, pp 933–940
- Vapnik V (1998) *Statistical learning theory*. Wiley, New York
- Villmann T, Haase S, Kaden M (2015) Kernelized vector quantization in gradient-descent learning. *Neurocomputing* 147:83–95
- Villmann T, Kaden M, Nebel D, Biehl M (2015) Learning vector quantization with adaptive cost-based outlier-rejection. In: Azzopardi G, Petkov N (eds) *Proceedings of 16th international conference on computer analysis of images and pattern, CAIP 2015, Valetta-Malta*, volume Part II of LNCS 9257. Springer, Berlin, Heidelberg, pp 772–782
- Villmann T, Kaden M, Bohnsack A, Saralajew S, Villmann J-M, Drogies T, Hammer B (2016) Self-adjusting reject options in prototype based classification. In: Merényi E, Mendenhall MJ, O'Driscoll P (eds) *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. Springer, Berlin, Heidelberg, pp 269–279

- Villmann T, Schleif F-M, Kaden M, Lange M (eds) (2014) Advances in self-organizing maps and learning vector quantization - proceedings of the 10th international workshop, WSOM 2014, Mittweida. Number 295 in Advances in intelligent systems and computing. Springer, Heidelberg
- Wilcoxon F (1945) Andividual comparisons by ranking methods. *Biometrics* 1:80–83
- Yan L, Dodier R, Mozer MC, Wolniewicz R (2003) Optimizing classifier performance via approximation to the Wilcoxon–Mann–Witney statistics. In: Proceedings of the 20th international conference on machine learning. AAAI Press, Menlo Park, pp 848–855
- Yu G, Russell W, Schwartz R, Makhoul J (1990) Discriminant analysis and supervised vector quantization for continuous speech recognition. In: ICASSP-90, international conference on acoustics, speech and signal processing, volume II, pp 685–688, Piscataway. IEEE, IEEE Service Center