

Title:

On the generalization ability of GRLVQ networks

article type:

regular paper submitted to NPL, NEPL64, revised

authors:

Barbara Hammer, Marc Strickert, LNM, Department of Mathematics/Computer Science, Universität Osnabrück, Germany,
and

Thomas Villmann, Clinique for Psychotherapy, Universität Leipzig, Germany,

corresponding author:

address: Barbara Hammer,
Department of Mathematics/Computer Science,
Universität Osnabrück,
Albrechtstr. 28,
Germany

e-mail: hammer@informatik.uni-osnabrueck.de

phone: +49 (0)541 969-2488

fax: +49 (0)541 969-2770

This paper has not been submitted elsewhere in identical or similar form, nor will it be during the first three months after its submission to Neural Processing Letters.

On the generalization ability of GRLVQ networks

Barbara Hammer and Marc Strickert

LNM, Department of Mathematics/Computer Science, Universität Osnabrück, Germany, e-mail: {hammer, marc}@informatik.uni-osnabrueck.de

Thomas Villmann

Clinique for Psychotherapy, Universität Leipzig, Germany, e-mail: villmann@informatik.uni-leipzig.de

April 14, 2004

Abstract. We derive a generalization bound for prototype-based classifiers with adaptive metric. The bound depends on the margin of the classifier and is independent of the dimensionality of the data. It holds for classifiers based on the Euclidean metric extended by adaptive relevance terms. In particular, the result holds for relevance learning vector quantization (RLVQ) [4] and generalized relevance learning vector quantization (GRLVQ) [19].

Keywords: generalization bounds, LVQ, relevance LVQ, margin optimization, adaptive metric

1. Introduction

Learning Vector Quantization (LVQ) as proposed by Kohonen and alternative prototype-based classifiers constitute intuitive and powerful learning algorithms [6, 23, 24]. They show excellent generalization for unseen data for high-dimensional inputs. A theoretical explanation for this fact has recently been established [9]: LVQ generalization bounds have been derived, which do not depend on the input dimensionality but only on the so-called hypothesis margin. Thus LVQ networks can be interpreted as learning algorithms which aim at structural risk minimization comparable to support vector machines (SVMs) [8]. LVQ provides solutions in terms of prototypical vectors, whereas SVM leads to classifiers expressed in terms of support vectors which are points at the borders of the classification constraints.

Prototype-based classifiers crucially depend on the representation of the data and on the metric structure of the input space. Usually, the



© 2004 Kluwer Academic Publishers. Printed in the Netherlands.

Euclidean metric is chosen. LVQ and variants fail if this metric is not appropriate for the given data and the learning task. High-dimensional and heterogeneous data cause a problem for the learning: irrelevant dimensions and accumulated noise can easily disrupt relevant information for classification. To overcome this problem, several algorithms which substitute the standard Euclidean metric by an adaptive version have been proposed: the algorithms of Gustafson and Kessel or Gath and Geva extend the unsupervised prototype-based k-means clustering algorithm to an adaptive metric described by a full matrix [3, 11, 15]. Further modifications of classical k-means use more complex cluster shapes such as fuzzy-k-varieties or fuzzy-k-shell [10]. Many statistical approaches offer a uniform method to determine the relevance of the inputs, see e.g. [12, 13, 36]. Several methods extend the unsupervised self-organizing map and variants to alternative metrics such as [14, 25]. Metric adaptation of SOM can be based on auxiliary information or class labels as recently proposed by Kaski and, earlier, by Cherkassky, Mulier and coworkers [7, 21, 22, 27]. Approaches which equip supervised LVQ with an adaptive diagonal metric have also been presented [4, 19, 30]. Particularly, generalized relevance learning vector quantization (GRLVQ) as introduced in [19] has proven to be successful in various application areas such as processing satellite images or time series prediction [17, 35, 41]. Thus, the choice of a correct metric is an important issue of prototype-based approaches.

Another issue of prototype-based classifiers is the stability of the learning procedure. Basic LVQ tends to be unstable for overlapping classes. Variants like LVQ2.1 and LVQ3 as proposed by Kohonen restrict the update dynamics to a local window to partially prevent the divergence, however, a correct parameter choice and training is still tricky and only based on heuristics [24]. The question whether one can design an appropriate cost function for variants of LVQ, which is optimized during training and thus characterizes its dynamic behavior,

is closely related to the design of stable variants of LVQ. The earliest proposal of a cost function of LVQ can be found in [32]. As discussed in the article [32], basic LVQ and LVQ2.1 obey cost functions which are discontinuous at the borders of the receptive fields causing instability of these methods for overlapping classes. The authors also propose an alternative formulation which yields a learning rule quite similar to LVQ2.1, generalized LVQ (GLVQ) [32]. As shown in [33], GLVQ prevents instabilities; it is explicitly computed in [17] that the derivatives of this cost function are well defined also for the class borders. A recent alternative cost function has been proposed in [34], which gives as a limit case another learning rule quite similar to LVQ. As pointed out in [40], the cost function can be extended to metric adaptation involving relevance updates comparable to GRLVQ. We would like to mention that the discussion of a cost function of LVQ is related to the discussion of a cost function for the unsupervised self-organizing map, see e.g. [20].

Extensions of LVQ, which are derived from a cost function or which incorporate alternative, possibly adaptive metrics, are only slightly more complex than original LVQ. These learning algorithms share the intuitive Hebbian update rules and the representation of the classifier in terms of characteristic prototypes. Some extensions of LVQ, however, provide much better classification accuracy or stability than the original version, see e.g. [4, 19]. In particular, a problem-adjustable metric in combination with LVQ can adequately deal with high-dimensional and heterogeneous data. The extension GRLVQ, which includes an adaptive diagonal metric and the cost function of GLVQ, even achieved a classification accuracy competitive to the SVM in a large scale problem from computational biology [18]. The question occurs, which improvements of LVQ can also be supported by a theoretical counterpart to accompany the empirical findings of previous work. The theoretical large margin bound on the generalization ability of LVQ, as derived in [9], does not transfer to the case of an adaptive metrics. Moreover, it

is not clear, which cost function shall be preferred with respect to the generalization ability of the classifier. Hence, no theoretical justification for the excellent generalization ability of some of these extended LVQ networks with adaptive metric is available so far.

The aim of this paper is to give a theoretical guarantee that extensions of simple LVQ by adaptive diagonal metrics possess similar generalization ability as standard LVQ; additionally, they are more flexible due to their adaptive metric parameters. The bounds which we get are dimensionality independent large margin bounds, and we will derive a connection of the cost function of GRLVQ to large margin optimization. The cost function of GLVQ and GRLVQ has been mainly designed to achieve greater stability. Thus this connection to large margin optimization is interesting since it shows that, as a side effect, this formulation aims at structural risk minimization during training.

The paper is structured as follows: first, we will formally introduce LVQ classifiers and some extensions. Afterwards, we derive a large margin bound for LVQ classifiers with adaptive diagonal metric. This bound holds for the function class given by prototype-based classifiers and adaptive metric, independent of the particular training algorithm. We will point out, however, that the specific training algorithms GLVQ and GRLVQ include the margin as a term of their respective cost function. Hence, these specific variants can be interpreted as margin maximization algorithms. Finally, we discuss in which aspects our findings extend previous work, what the practical consequences of these theoretical bounds are, and what the relations between LVQ-type learning and SVM are.

2. Generalized learning vector quantization

Assume that a finite training set $\{(x^i, y_i) \in \mathbb{R}^n \times \{1, \dots, C\} \mid i = 1, \dots, m\}$ is given. Classes are enumerated by $1, \dots, C$ and \mathbb{R}^n denotes

the potentially high-dimensional data space. Denote by $X = \{x^i \mid i = 1, \dots, m\}$ all input signals of the training set. Components of a vector $x \in \mathbb{R}^n$ are referred to by subscripts, i.e., $x = (x_1, \dots, x_n)$. Learning vector quantization (LVQ), as introduced by Kohonen [23], represents every class c by a set $W(c)$ of weight vectors (prototypes) in \mathbb{R}^n . Weight vectors are denoted by w^r and their respective class label is referred to by c_r . A new signal $x \in \mathbb{R}^n$ is classified by the winner-takes-all rule of an LVQ network, i.e.

$$x \mapsto c(x) = c_r \text{ such that } d(x, w^r) \text{ is minimum.} \quad (1)$$

$d(x, w^r) = \|x - w^r\|^2 = \sum_{i=1}^n (x_i - w_i^r)^2$ denotes the squared Euclidean distance of the data point x to the prototype w^r . The respective closest prototype w^r is called winner or best matching unit. The subset

$$\Omega_r = \{x^i \in X \mid d(x^i, w^r) \text{ is minimum}\}$$

is called receptive field of neuron w^r .

The training algorithm of LVQ aims at minimizing the classification error on the given training set. I.e., the difference between the set of points belonging to the c th class, $\{x^i \in X \mid y_i = c\}$ and the receptive fields of the corresponding prototypes, $\bigcup_{w^r \in W(c)} \Omega_r$, is minimized by the adaptation process. Training iteratively presents randomly chosen data from the training set and adapts the closest prototype by Hebbian learning in the following way: if a vector x^i is presented, the update rule for the winner w^r has the form

$$\Delta w^r = \begin{cases} \epsilon \cdot (x^i - w^r) & \text{if } c^r = c(x^i) \\ -\epsilon \cdot (x^i - w^r) & \text{otherwise.} \end{cases}$$

$\epsilon \in (0, 1)$ is an appropriate learning rate. As explained in [32], this update can be interpreted as a stochastic gradient descent on the cost function

$$\text{Cost}_{\text{LVQ}} = \sum_{x^i \in X} f_{\text{LVQ}}(d_{r+}, d_{r-}).$$

d_{r+} denotes the squared Euclidean distance of x^i to the closest prototype w^{r+} labeled with $c_{r+} = y_i$, and d_{r-} denotes the squared Euclidean distance to the closest prototype w^{r-} labeled with a label c_{r-} different from y_i . For standard LVQ, the function is

$$f_{\text{LVQ}}(d_{r+}, d_{r-}) = \begin{cases} d_{r+} & \text{if } d_{r+} \leq d_{r-} \\ -d_{r-} & \text{otherwise} \end{cases}$$

Obviously, this cost function is highly discontinuous, and instabilities arise for overlapping data distributions.

Various alternatives have been proposed which substitute the training rule of LVQ by another in order to achieve more stable training in case of overlapping classes or noisy data. Kohonen's LVQ2.1 optimizes the cost function which is obtained by setting in the above sum $f_{\text{LVQ2.1}}(d_{r+}, d_{r-}) = I_w(d_{r+} - d_{r-})$, whereby I_w yields the identity inside a window where LVQ2.1 adaptation takes place, and I_w vanishes outside. Still this choice might produce an instable dynamic, and the window where adaptation takes place must be chosen carefully. Generalized LVQ (GLVQ) constitutes a stable alternative to LVQ2.1 [32]. The respective cost function can be obtained by setting

$$f_{\text{GLVQ}}(d_{r+}, d_{r-}) = \text{sgd} \left(\frac{d_{r+} - d_{r-}}{d_{r+} + d_{r-}} \right)$$

whereby $\text{sgd}(x) = (1 + \exp(-x))^{-1}$ denotes the logistic function. As discussed in [33], the additional scaling factors avoid numerical instabilities and divergent behavior. We omit the update formulas for the prototypes for LVQ2.1 and GLVQ, which are obtained taking the derivative [24, 32].

Recently, it has been shown that the term

$$(\|x^i - w^{r-}\| - \|x^i - w^{r+}\|)/2$$

constitutes the so-called hypothesis margin of a prototype-based classifier according to the winner-takes-all rule (1). The hypothesis margin refers to the distance in an appropriate norm, which the classifier can

alter without changing the classification. Generalization bounds which depend on this hypothesis margin have been derived in [9]. Note that LVQ2.1 and GLVQ express this margin in terms of their cost functions, hence, they can be interpreted as margin optimization learning algorithms.

The winner-takes-all classification (1) used by the above networks crucially depend on the Euclidean metric, which may lead to inaccurate results for high-dimensional or heterogeneous data. A very simple and powerful extension introduces relevance terms which weight the input dimensions appropriately, i.e. the squared Euclidean metric $d(x, y) = \|x - y\|^2$ is substituted by the weighted metric,

$$d^\lambda(x, y) = \sum_{i=1}^n \lambda_i (x_i - y_i)^2$$

whereby $\lambda_i \geq 0$ and the relevance vector is normalized to $\sum_i \lambda_i = 1$. The classification of a data point x is given by the modified winner-takes-all rule

$$x \mapsto c(x) = c_r \text{ such that } d^\lambda(x, w^r) \text{ is minimum.} \quad (2)$$

Appropriate scaling terms λ_i allow to choose small relevance factors for less relevant or noisy dimensions. Since appropriate relevances are not a priori known, they are also adapted during training. Relevance learning vector quantization (RLVQ), as introduced in [4], constitutes a stochastic gradient descent on the cost function corresponding to simple LVQ

$$\text{Cost}_{\text{RLVQ}} = \sum_{x^i \in X} f_{\text{LVQ}}(d_{r^+}^\lambda, d_{r^-}^\lambda)$$

whereby $d_{r^+}^\lambda$ and $d_{r^-}^\lambda$ refer to distance of the closest prototypes from x^i now measured with respect to the squared *weighted* Euclidean norm with the same or a different class label. Generalized relevance learning vector quantization minimizes the cost function which is obtained for

$$f_{\text{GLVQ}}(d_{r^+}^\lambda, d_{r^-}^\lambda) = \text{sgd} \left(\frac{d_{r^+}^\lambda - d_{r^-}^\lambda}{d_{r^+}^\lambda + d_{r^-}^\lambda} \right)$$

depending on the *weighted* Euclidean distances d_{r+}^λ and d_{r-}^λ . Again, we omit the update formulas for prototypes and relevance terms for RLVQ and GRLVQ, which are obtained taking the derivatives of the above functions [4, 19].

The generalization bound derived in the article [9] holds for all prototype-based classifiers for which the classification rule is given by (1). The bound does no longer hold for prototype-based classifiers with classification rule (2) exhibiting changing parameters λ during training. We will derive a large margin generalization bound for the latter case in the following.

3. Generalization bounds

The generalization ability of a classifier refers to the comparison of the training error with the expected error for new data. There are various ways to formalize and prove the generalization ability of classifiers, such as the popular VC-theory [39] or recent argumentation based on Rademacher and Gaussian complexity [2]. Here, we consider the situation of binary classification problems, i.e. only two classes are given, and we assume the classes are labeled 1 and -1 . Assume an (unknown) probability measure P is given on $\mathbb{R}^n \times \{-1, 1\}$. Training samples (x^i, y_i) are drawn independently and identically distributed (i.i.d. for short) from $\mathbb{R}^n \times \{-1, 1\}$. P^m refers to the product of P if m examples $(x^1, y_1), \dots, (x^m, y_m)$ are chosen. The unknown regularity shall be learned by a GRLVQ-network or some other prototype-based classifier with adaptive diagonal metric. The classifier is characterized by its set of prototypes w^1, \dots, w^p in \mathbb{R}^n (p denoting the number of prototypes) and the relevance terms $\lambda_1, \dots, \lambda_n$ which describe the weighted metric. The function computed by the classifier is given by

the winner-takes-all rule defined in (2). Denote by

$$\mathcal{F} = \{f : \mathbb{R}^n \rightarrow \{-1, 1\} \mid f \text{ is given by (2) depending on } w^1, \dots, w^p, \lambda \in \mathbb{R}^n\}$$

the class of functions which can be computed by such a network. The goal of learning is to find a function $f \in \mathcal{F}$ for which the probability

$$E_P(f) := P(y \neq f(x))$$

is minimum. Since the underlying regularity P is not known and only examples (x^i, y_i) are available for characterizing this regularity, training tries to minimize the empirical training error

$$\hat{E}_m(f) := \sum_{i=1}^m 1_{y_i \neq f(x^i)} / m$$

whereby $1_{y_i \neq f(x^i)}$ indicates whether x^i is mapped to the desired class y_i or not. Generalization means that $\hat{E}_m(f)$ is representative for $E(f)$ with high probability if the examples are chosen according to P^m such that optimization of the empirical training error will eventually approximate the underlying regularity.

Due to the chosen cost function, GRLVQ minimizes the training error and, in addition, also optimizes the margin of the classifier during training. Given a point x with desired output y , we define the margin as the value

$$M_f(x, y) := -d_{r+}^\lambda + d_{r-}^\lambda$$

whereby d_{r+}^λ refers to the squared weighted distance of the closest prototype of the same class as x , and d_{r-}^λ refers to the squared weighted distance of the closest prototype labeled with a different class from x . x is classified incorrectly iff $M_f(x, y)$ is negative. Otherwise, x is classified correctly with ‘security’ margin $M_f(x, y)$. Due to the choice of the cost function of GRLVQ which involves this term within the denominator, GRLVQ aims at maximizing this margin. Following the approach [2]

we define the loss function

$$L : \mathbb{R} \rightarrow \mathbb{R}, t \mapsto \begin{cases} 1 & \text{if } t \leq 0 \\ 1 - t/\rho & \text{if } 0 < t \leq \rho \\ 0 & \text{otherwise} \end{cases}$$

for fixed $\rho > 0$. The term

$$\hat{E}_m^L(f) := \sum_{i=1}^m L(M_f(x^i, y_i))/m$$

accumulates the number of errors made by f and, in addition, punishes all correctly classified points, if their margin is smaller than ρ .

We will show now that this modified empirical error, which also includes the margin, is representative for the true error with high probability, whereby a bound independent from dimensionality is obtained. We assume that the support of the probability measure P is bounded, i.e. that for all data points x the inequality

$$\|x\| \leq B$$

holds for some $B > 0$, $\|\cdot\|$ denoting the standard Euclidean metric. In addition, all prototypes w are restricted by

$$\|w\| \leq B.$$

According to [2](Theorem 7) we can estimate for all $f \in \mathcal{F}$ with probability at least $1 - \delta/2$

$$E_P(f) \leq \hat{E}_m^L(f) + \frac{2K}{\rho} \cdot G_m(\mathcal{F}) + \sqrt{\frac{\ln(4/\delta)}{2m}}$$

whereby K is a universal positive constant and $G_m(\mathcal{F})$ is the so-called Gaussian complexity of the considered function class which we now define. The empirical Gaussian complexity is given by

$$\hat{G}_m(\mathcal{F}) = E_{g_1, \dots, g_m} \left(\sup_{f \in \mathcal{F}} \left| \frac{2}{m} \sum_{i=1}^m g_i \cdot f(x^i) \right| \right)$$

for which expectation is taken with respect to independent Gaussian variables g_1, \dots, g_m with zero mean and unit variance. The Gaussian

complexity is the expectation over the i.i.d. points x^i according to the marginal distribution induced by P : $G_m(\mathcal{F}) = E_{x^1, \dots, x^m} \hat{G}_m(\mathcal{F})$. Both complexities measure the richness of the function class \mathcal{F} and constitute convenient alternatives to the standard VC-dimension which can also be estimated for prototype-based classifiers.

The classification given by the winner-takes-all rule (2) can be reformulated as fixed Boolean formula over terms of the form $d_{r^i}^\lambda - d_{r^j}^\lambda$ with $d_{r^i}^\lambda$ and $d_{r^j}^\lambda$ constituting the weighted squared Euclidean distance of a given input x to two prototypes w^i and w^j with different class labels. Note that the number of such terms is upper bounded by $p \cdot (p-1)/2$ if p prototypes are available within the classifier. According to [2](Theorem 16) we find

$$G_m(\mathcal{F}) \leq p \cdot (p-1) \cdot G_m(\mathcal{F}_{ij})$$

whereby \mathcal{F}_{ij} denotes the restricted class of classifiers which can be implemented with only two prototypes w^i and w^j with different class label. Denote by Λ the diagonal matrix of relevance terms. Then we find

$$\begin{aligned} d_{r^i}^\lambda - d_{r^j}^\lambda &\leq 0 \\ \iff (x - w^{r^i})^t \cdot \Lambda \cdot (x - w^{r^i}) - (x - w^{r^j})^t \cdot \Lambda \cdot (x - w^{r^j}) &\leq 0 \\ \iff 2 \cdot (\Lambda \cdot w^{r^i} - \Lambda \cdot w^{r^j})^t x^i + (w^{r^j})^t \cdot \Lambda \cdot w^{r^j} - (w^{r^i})^t \cdot \Lambda \cdot w^{r^i} &\geq 0 \end{aligned}$$

Hence, we can embed \mathcal{F}_{ij} into the class of functions implemented by a simple perceptron or a linear classifier. Since $\|x\| \leq B$, the length of inputs to the linear classifier can be restricted by $B+1$ (including the bias term). Since all prototypes w are restricted by $\|w\| \leq B$ and the relevance terms add up to 1, the size of the weights of the linear classifier is restricted by $4B + 2B^2$. The empirical Gaussian complexity of this class of linear classifiers can be estimated according to [2](Lemma 22) by

$$\frac{4 \cdot B \cdot (B+1) \cdot (B+2) \cdot \sqrt{m}}{m}.$$

Since the empirical Gaussian complexity and the Gaussian complexity differ by more than ϵ with probability at most $2 \cdot \exp(-\epsilon^2 m/8)$ according

to [2](Theorem 11), we can estimate

$$E_P(f) \leq \hat{E}_m^L(f) + \frac{8K \cdot p(p-1)B(B+1)(B+2)\sqrt{m}}{\rho \cdot m} + \left(1 + \frac{8K \cdot p(p-1)}{\rho}\right) \sqrt{\frac{\ln 4/\delta}{2m}}$$

with probability of at least $1 - \delta$. This term limits the generalization error for all classifiers of the form (2) with adaptive metric if only two classes are dealt with and inputs and weights are restricted by B .

Note that this bound is independent of the dimensionality n of the data. It scales inversely to the margin ρ , i.e. the larger the margin the better the generalization ability. This generalization bound holds for all prototype-based classifiers which implement the winner-takes-all rule (2) with possibly adaptive relevance terms λ . In particular, it holds for the functions obtained by the algorithms RLVQ and GRLVQ. In addition, the bound indicates that GRLVQ includes the objective of structural risk minimization during training because a term which describes the margin is directly contained in the cost function of GRLVQ.

4. Discussion

We have derived a large margin bound on the generalization ability of prototype-based classifiers including adaptive metric parameters. The achieved bound is comparable to the bound obtained in [9] for simple LVQ. On the one hand side, this argumentation proposes a theoretical foundation for the – in practice highly efficient though simple – extension of standard LVQ to an adaptive diagonal metric that is used in RLVQ and GRLVQ. On the other hand, the argumentation indicates that the cost function of GRLVQ leads, besides the improved stability and robustness of the algorithm, to structural risk optimization during training.

We would like to discuss a few points in more detail. The work [9] derives large margin bounds for prototype-based classifiers, in partic-

ular for LVQ-type networks. However, the bounds hold for the Euclidean metric and do not cover the case of adaptive metric parameters. There exist related discussions in the SVM-community about how SVM-kernels can be adapted according to given data, and for which adaptations theoretical generalization bounds still hold, see e.g. [5]. Apart from the work [9], bounds on the generalization ability of prototype-based classifiers have already previously been derived based on estimations of the growth function, see e.g. [31]. These bounds, however, assume the prototypes to be fixed. Thus, they do not apply to LVQ-type learning for which the prototype position is optimized based on the given labeled training data. One strength of LVQ and the discussed variants is that the prototypes and metric parameters are determined during training to give optimum classification accuracy on the training set. The bounds derived above apply to this case.

We have shown that an adaptive diagonal metric does not increase the bounds on the generalization error, although the flexibility of the classifier is extended. For more general metrics such as a full matrix the generalization bounds are not clear. Thus, if a diagonal metric is used, the classification accuracy can increase significantly and one remains on the ‘safe side’ with respect to the generalization ability. In addition, we have shown that the term $-d_{r+}^{\lambda} + d_{r-}^{\lambda}$ can be interpreted as margin and plays an important role in the generalization ability of the classifier. Thus cost functions which include this term in the formulation also optimize the margin during training. For alternative cost functions such as [34], this is not clear. As a consequence, the combination of the cost function of GLVQ [32] and relevance learning seems a good idea from the point of learning theory. This supports the experimental finding that GRLVQ often yields accurate and efficient classification [17, 18, 19, 35, 41].

In principle, tight bounds from learning theory on the generalization error could also be valuable for model selection and tuning of hyper-

parameters such as learning rate, etc. So far, hyperparameter tuning is often done by crossvalidation, i.e. the average test error on a hold-out set serves as estimation of the generalization error of a fixed setting. The hyperparameters with lowest estimated generalization error are then chosen. Learning theory offers alternative bounds on the generalization error of the form

$$E_P(f) \leq \hat{E}_m(f) + \text{estimation of the structural risk } (*),$$

which do not require a hold-out set and which are less sensitive to the actual data distribution in the training set. A successful application of this principle has been proposed e.g. in [31]. However, too loose bounds on the structural risk might be misleading and the method is not applicable then. Alternative proposals from learning theory closer to the actual distribution of the data such as [1] or simple cross-validation might yield more reliable results in such cases.

Here, we obtain a result of the form (*) for GRLVQ networks. However, since the focus of this work is the principled theoretical support of existing models, an empirical test of the tightness of this bound is beyond the scope of this article. Note that it would be necessary to substitute the universal constant involved in the bound in this case.

Besides LVQ, alternative large margin classifiers exist. Among them, SVM is supposed to be one of the best classification and regression models available today and, unlike LVQ variants, it is directly based on the principle of structural risk minimization in a sound mathematical way [38]. However, the experiences from the StatLog project clearly indicate that there is no single best algorithm for every type of classification problem and the method of choice highly depends on the given circumstances [28]. Also for simple problems, alternative methods might give better, simpler, or more efficient solutions than other particularly popular ones, demonstrated e.g. for SVM and the set covering machine

in [26]. Thus, it is worthwhile to develop and investigate alternatives to SVM.

LVQ-type classifiers have been designed as simple and intuitive classification methods, and variants show quite good performance in practice [17, 18, 24]. Since they have not explicitly been designed for large margin optimization, it is interesting to see that large margin bounds can be derived also for these models. Good generalization ability can be expected for these classifiers, because some training methods involve large margin optimization related to the associated cost function.

What could be benefits of prototype-based methods compared to SVM? Unlike SVM, LVQ expresses a solution in terms of prototypes, i.e. typical positions within the data space. SVM expands solutions in terms of support vectors which are positions at the borders of the constraints given by atypical positions or wrongly classified points. Depending on the area of application, an expansion in terms of atypical vectors might not be the best solution. On the one hand side, a solution in terms of prototypes can be more natural for a human investigator, and GRLVQ and variants allow to directly extract an approximation of their behavior in terms of a decision tree [16]. Thus prototype-based classifiers are interesting if further insight into the classifier is necessary for some reason, e.g. a human-understandable explanation of the classification is desired in decision support systems. On the other hand, if large data sets are dealt with, solutions in terms of prototypes are often sparser than solutions provided by SVM. The number of support vectors is typically a fraction of the training set size and large datasets yield quite complex solutions. The size of LVQ networks depends on the number of modes in the underlying data distribution and it can be controlled independent of the training set size. In practice, GRLVQ training time scales linearly with respect to the number of data points. The classifier size and the classification time is constant with respect to the training set size. This has been demonstrated in comparison to SVM

results for a large benchmark problem in computational biology, for which GRLVQ yields solutions with the same accuracy as a comparable SVM, but of which the size is up to a factor 100 smaller depending on the training set size [18].

Naturally, we do not claim that GRLVQ is the best solution for such cases and interesting alternatives which combine SVM principles with prototype-based learning like the relevance vector machine exist [37]. However, GRLVQ constitutes a simple, intuitive, and efficient algorithm with theoretical guarantees as shown in this article, which might be worth trying.

References

1. D. Anguita, S. Ridella, F. Riviaccio, and R. Zunino. Automatic hyperparameter tuning for support vector machines. In: J.R.Dorronsoro (ed.) *ICANN 2002*, Springer, 1345-1350, 2002.
2. P.L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: risk bounds and structural results. *Journal of Machine Learning and Research* 3:463-482, 2002.
3. J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.
4. T. Bojer, B. Hammer, D. Schunk, and K. Tluk von Toschanowitz. Relevance determination in learning vector quantization. In *Proc. of European Symposium on Artificial Neural Networks (ESANN'01)*, pages 271-276, Brussels, Belgium, 2001. D facto publications.
5. O. Bousquet and D.J.L. Herrmann. On the complexity of learning the kernel matrix. In: *Advances in Neural Information Processing Systems 2002*, to appear.
6. Neural Networks Research Centre, Otaniemi: Helsinki University of Technology. *Bibliography on the self-organizing map (SOM) and learning vector quantization (LVQ)*. Available at: <http://iinwww.ira.uka.de/bibliography/Neural/SOM.LVQ.html>
7. V. Cherkassky, D. Gehring, and F. Mulier. Comparison of adaptive methods for function estimation from samples. *IEEE Transactions on Neural Networks*, 7:969-984, 1996.
8. C. Cortes and V. Vapnik. Support vector network. *Machine Learning*, 20 (1995), 1-20.
9. K. Crammer, R. Gilad-Bachrach, A. Navot, and A. Tishby. Margin analysis of the LVQ algorithm. In: *Advances in Neural Information Processing Systems 2002*, to appear.
10. R.N. Davé. Fuzzy shell-clustering and application to circle detection in digital images. *International Journal of General Systems* 16:343-355, 1990.

11. I. Gath and A.B. Geva. Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11:773-781, 1989.
12. T. van Gestel, J. A. K. Suykens, B. de Moor, and J. Vandewalle. Automatic relevance determination for least squares support vector machine classifiers. In M. Verleysen, editor, *European Symposium on Artificial Neural Networks*, 13–18, 2001.
13. Y. Grandvalet. Anisotropic noise injection for input variables relevance determination. *IEEE Transactions on Neural Networks*, 11(6):1201–1212, 2000.
14. S. Günter and H. Bunke. Self-organizing map for clustering in the graph domain. *Pattern Recognition Letters*, 23:401–417, 2002.
15. E.E. Gustafson and W.C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In: *IEEE CDC*, pages 761-766, San Diego, California, 1979.
16. B. Hammer, A. Rechten, M. Strickert, T. Villmann, Rule extraction from self-organizing networks. In: J. R. Dorronsoro (ed.), *ICANN 2002*, Springer, 877-882, 2002.
17. B. Hammer, M. Strickert, and T. Villmann. Supervised neural gas with general similarity measure. To appear in *Neural Processing Letters*.
18. B. Hammer, M. Strickert, and T. Villmann. Prototype recognition of splice sites. In: U.Seiffert, L.C.Jain, and P.Schweitzer (eds.), *Bioinformatics using Computational Intelligence Paradigms*, Springer, 2004, in press.
19. B. Hammer and T. Villmann. Generalized relevance learning vector quantization. *Neural Networks* 15:1059-1068, 2002.
20. T. Heskes. Energy functions for self-organizing maps. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 303–315. Springer, 1999.
21. S. Kaski and J. Sinkkonen. A topography-preserving latent variable model with learning metrics. In: N. Allinson, H. Yin, L. Allinson, and J. Slack, editors, *Advances in Self-Organizing Maps*, pages 224–229, Springer, 2001.
22. S. Kaski. Bankruptcy analysis with self-organizing maps in learning metrics. *IEEE Transactions on Neural Networks* 12:936-947, 2001.
23. T. Kohonen. Learning vector quantization. In M. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 537–540. MIT Press, 1995.
24. T. Kohonen. *Self-Organizing Maps*. Springer, 1997.
25. T. Kohonen and P. Somervuo. How to make large self-organizing maps for nonvectorial data. *Neural Networks* 15(8-9):945-952, 2002.
26. M. Marchand and J. Shawe-Taylor. The set covering machine. *Journal of Machine Learning Research* 3:723-746, 2002.
27. F. Mulier. *Statistical Analysis of Self-Organization*. Ph.D. Thesis, University of Minnesota, Minneapolis, 1994.
28. D. Michie, D.J. Spiegelhalter, and C.C. Taylor (eds.) *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994.
29. G. Patané and M. Russo. The enhanced LBG algorithm. *Neural Networks* 14:1219-1237, 2001.
30. M. Pregoner, G. Pfurtscheller, and D. Flotzinger. Automated feature selection with distinction sensitive learning vector quantization. *Neurocomputing* 11:19-29, 1996.
31. S. Ridella, S. Rovetta, and R. Zunino. K-winner machines for pattern classification. *IEEE Transactions on Neural Networks* 12(2):371-385, 2001.
32. A.S. Sato and K. Yamada. Generalized learning vector quantization. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 423–429, MIT Press, 1995.

33. A.S. Sato and K. Yamada. An analysis of convergence in generalized LVQ. In L. Niklasson, M. Bodén, and T. Ziemke (eds.) *ICANN'98*, pages 172-176, Springer, 1998.
34. S. Seo and K. Obermeyer. Soft learning vector quantization. *Neural computation* 15:1589-1604, 2003.
35. M. Strickert, T. Bojer, and B. Hammer. Generalized relevance LVQ for time series. In: G.Dorffner, H.Bischof, K.Hornik (eds.), *Artificial Neural Networks - ICANN'2001*, Springer, pages 677-683, 2001.
36. R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58, 267-288, 1996.
37. M.E. Tipping. The relevance vector machine. *Journal of Machine Learning research* 1:211-244, 2001.
38. V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.
39. V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications* 16(2):264-280, 1971.
40. T. Villmann and B. Hammer. *Metric adaptation and relevance learning in learning vector quantization*. Technical Report, Reihe P, Heft 247, FB Mathematik/Informatik, Universität Osnabrück, 2003.
41. T. Villmann, E. Merenyi, and B. Hammer. Neural maps in remote sensing image analysis. *Neural Networks* 16(3-4): 389-403, 2003.

