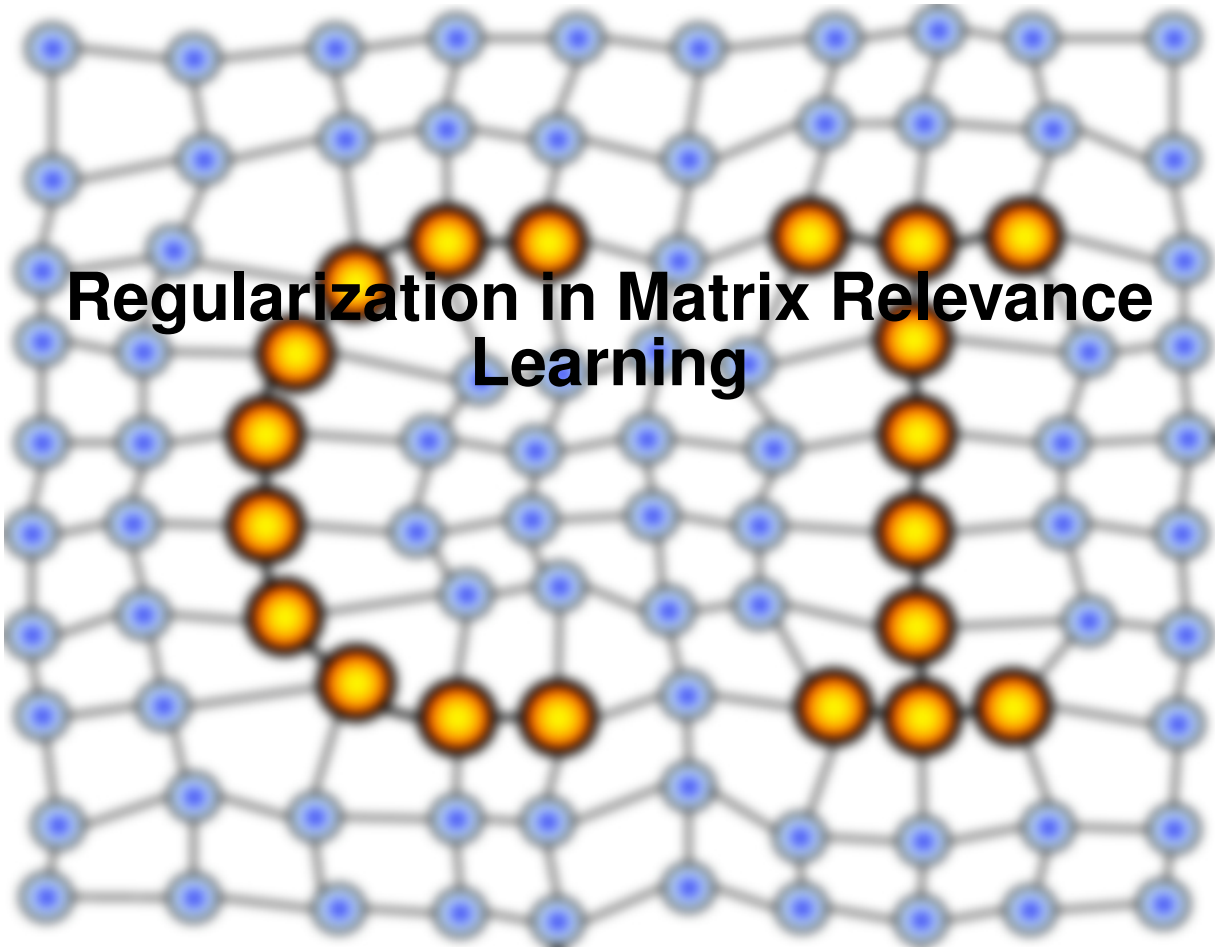
MACHINE LEARNING REPORTS



Regularization in Matrix Relevance Learning

Report 02/2008
Submitted: 23.10.2008
Published: 30.10.2008

Petra Schneider[1] and Kerstin Bunte[1] and Han Stiekema[1] and Barbara Hammer[2] and Thomas Villmann[3] and Michael Biehl[1]
(1) University of Groningen, Institute for Mathematics and Computing Science
P.O. Box 407, 9700 AK Groningen - The Netherlands
(2) Clausthal University of Technology, Institute of Computer Science
Julius Albert Strasse 4, 38678 Clausthal-Zellerfeld - Germany
(3) University of Leipzig, Department of Medicine
Semmelweisstrasse 10, 04103 Leipzig - Germany

Abstract

We present a regularization method which extends the recently introduced Generalized Matrix LVQ. This learning algorithm extends the concept of adaptive distance measures in LVQ to the use of relevance matrices. In general, relevance learning can display a tendency towards over-simplification in the course of training. An overly pronounced elimination of dimensions in feature space can have negative effects on the performance and may lead to instabilities in the training. Complementing the standard GMLVQ cost function by an appropriate regularization term prevents this unfavorable behavior and can help to improve the generalization ability. The approach is first tested and illustrated in terms of artificial model data. Furthermore we apply the scheme to a benchmark classification problem from the medical domain. For both data sets, we demonstrate the usefulness of regularization also in the case of rank limited relevance matrices, i.e. GMLVQ with an implicit, low dimensional representation of the data.

1 Introduction

Learning Vector Quantization (LVQ) as introduced by Kohonen is a particularly intuitive and simple though powerful classification scheme [Koh97, Hel02, BGH07] which is very appealing for several reasons: The method is easy to implement, the complexity of the resulting classifier can be controlled by the user, the classifier can naturally deal with multi-class problems. Unlike many alternative classification schemes such as feed-forward networks or the Support Vector Machine (SVM), LVQ system is straightforward to interpret because of the intuitive assignment of data to the class of the closest prototype. For these reasons, LVQ has been used in a variety of academic and commercial applications such as image analysis, bioinformatics, telecommunication, robotics, etc. Variants of LVQ which can be derived from an explicit cost function are particularly interesting. Several proposals for cost functions can be found in the literature, one example being Generalized LVQ (GLVQ) [SY96] which forms the basis for the method we will consider in this article. However, LVQ and variants often rely on the standard Euclidean metric which is not necessarily appropriate. This is the case, e.g., for high dimensional data where noise accumulates and likely corrupts the classification, for heterogeneous data where the importance and nature of the dimensions differs, and for data which involves correlations of the dimensions. In these cases, which are quite common in practice, simple LVQ may fail. So-called relevance learning techniques [BHST01, HV02, VSH06] aim to optimize the distance measure for the concrete application. Generalized Relevance LVQ (GRLVQ) [HV02], is a powerful alternative to GLVQ which extends the Euclidean distance with scaling or relevance factors for all features. The weight values are adapted to the data during training in parallel to the prototypes. The choice of this similarity measure has turned out particularly suitable in many practical applications since it can account for irrelevant or inadequately scaled dimensions. At the same time, it allows for straightforward interpretation of the result because the relevance profile can directly be interpreted as the contribution of the dimensions to the classification [HV02]. The recently introduced Generalized Matrix LVQ algorithm (GMLVQ) [SBH07a, SBH07b, BHS06] constitutes a further generalization of GRLVQ. The method uses of a full adaptive matrix of relevance factors in the distance measure which accounts for pairwise correlations of features. By means of an implicit linear transformation of the data, the algorithm yields a discriminative distance measure which is particularly suitable for the given classification task. While the flexibility of the method is widely extended by matrix adaptation, the excellent generalization ability of matrix LVQ can be guaranteed by means of large margin generalization bounds [SBH07a, SBH07b, BHS06].

However, metric adaptation techniques may be subject to over-simplification of the classifier as the algorithms possibly eliminate too many dimensions which makes it impossible to reach the best performance (see e.g. [BBL07]).

We develop a regularization scheme for GRLVQ and GMLVQ to prevent the algorithms from over-simplifying the distance measure. To this end, the original GLVQ cost function is extended by a penalty term which punishes distinct relevance profiles. We demonstrate the behavior of the method by means of an artificial data set and one real world application. It is also applied to GMLVQ with rank limited relevance matrices, i.e. an implicit low-dimensional representation of the data.

2 Review of Generalized Matrix LVQ

LVQ aims at parameterizing a classification scheme in terms of prototypes. Assume training data $(\vec{\xi}_i, y_i) \in \mathbb{R}^N \times \{1, \ldots, C\}$ are given, N denoting the data dimensionality and C the number of different classes. An LVQ network consists of a number of prototypes which are characterized by their location in the feature space $\vec{w}_i \in \mathbb{R}^N$ and their class label $c(\vec{w}_i) \in \{1, \ldots, C\}$. Classification takes place by a winner takes all scheme. For this purpose, a (possibly parameterized) similarity measure d^λ is defined in $\mathbb{R}^N$. Often, the standard Euclidean metric is chosen. A data point $\vec{\xi} \in \mathbb{R}^N$ is mapped to the class label $c(\vec{\xi}) = c(\vec{w}_i)$ of the prototype i for which $d^\lambda(\vec{w}_i, \vec{\xi}) \leq d^\lambda(\vec{w}_j, \vec{\xi})$ holds for every $j \neq i$ (breaking ties arbitrarily). Learning aims at determining weight locations for the prototypes such that the given training data are mapped to their corresponding class labels. A very flexible learning approach has been introduced in [HSV05]. It is derived as a minimization of the cost function

$$f = \sum_i \phi \left(\frac{d_J^\lambda - d_K^\lambda}{d_J^\lambda + d_K^\lambda} \right) \tag{1}$$

where ϕ is a monotonic function, e.g. the logistic function or the identity $\phi(x) = x$ which we use throughout the following, $d_J^\lambda = d^\lambda(\vec{w}_J, \vec{\xi}_i)$ is the distance of data point $\vec{\xi}_i$ from the closest prototype $\vec{w}_J$ with the same class label y_i, and $d_K^\lambda = d^\lambda(\vec{w}_K, \vec{\xi}_i)$ is the distance from the closest prototype $\vec{w}_K$ with any class label different from y_i. Taking derivatives with respect to the prototypes and metric parameters yields gradient based adaptation rules. Fixing the similarity measure as standard Euclidean metric yields GLVQ [SY96]. The squared *weighted* Euclidean metric $d^\lambda(\vec{w}, \vec{\xi}) = \sum_i \lambda_i (w_i - \xi_i)^2$ where $\lambda_i \geq 0$ and $\sum_i \lambda_i = 1$ constitutes a powerful alternative, Generalized Relevance LVQ [HV02]. It is particularly suitable for high dimensional data with input dimensions of different (but a priori unknown) relevance. In Generalized Matrix LVQ [SBH07b, SBH07a], a full matrix which can account for pair-wise correlations of the dimensions, is used. The metric has the form

$$d^\Lambda(\vec{w}, \vec{\xi}) = (\vec{\xi} - \vec{w})^T \Lambda \, (\vec{\xi} - \vec{w}) \tag{2}$$

where Λ is an $N \times N$ matrix. The above similarity measure only corresponds to a meaningful distance if Λ is positive (semi-) definite. We can achieve this by substituting $\Lambda = \Omega^T \Omega$. The matrix Ω can be chosen in several different forms:

(a) Quadratic and symmetric, i.e. $\Omega \in \mathbb{R}^{N \times N}$, $\Omega_{ij} = \Omega_{ji}$

(b) Quadratic and non-symmetric, i.e. $\Omega \in \mathbb{R}^{N \times N}$, $\Omega_{ij} \neq \Omega_{ji}$

(c) Rectangular and non-symmetric, i.e. $\Omega \in \mathbb{R}^{M \times N}$ with $M < N$

Obviously, the quadratic, non-symmetric alternative constitutes a special case of the rectangular matrix with $M = N$.
Depending on the shape of Ω, the computation of d^Λ in terms of Ω differs. For symmetric matrices Ω we set $\Lambda = \Omega\Omega$ and get

$$d_1^\Lambda(\vec{w}, \vec{\xi}) = \sum_{i,j,k}^{N} (\xi_i - w_i) \Omega_{ik} \Omega_{kj} (\xi_j - w_j) \tag{3}$$

A non-symmetric rectangular ($M \neq N$) or quadratic ($M = N$) matrix Ω results in

$$d_2^\Lambda(\vec{w}, \vec{\xi}) = \sum_{i,j}^{N} \sum_{k}^{M} (\xi_i - w_i)\Omega_{ki}\Omega_{kj}(\xi_j - w_j) \tag{4}$$

To obtain the update rules, the derivatives of (2) with respect to $\vec{w}$ and Ω have to be computed. The derivative with respect to $\vec{w}$ reads

$$\frac{\partial d^\Lambda(\vec{w}, \vec{\xi})}{\partial \vec{w}} = -2\,\Lambda\,(\vec{\xi} - \vec{w}) = -2\,\Omega^T\Omega\,(\vec{\xi} - \vec{w}) \tag{5}$$

The different alternatives to formulate d^Λ in terms of Ω (Eq. (3) and Eq. (4)) lead us to two different derivatives of the distance measure with respect to a single metric parameter Ω_{lm}

$$\frac{\partial d_1^\Lambda(\vec{w}, \vec{\xi})}{\partial \Omega_{lm}} = \sum_j (\xi_l - w_l)\Omega_{mj}(\xi_j - w_j) + \sum_i (\xi_i - w_i)\Omega_{il}(\xi_m - w_m) \tag{6}$$

$$\frac{\partial d_2^\Lambda(\vec{w}, \vec{\xi})}{\partial \Omega_{lm}} = 2\sum_i (\xi_i - w_i)\Omega_{li}(\xi_m - w_m) \tag{7}$$

Using Eq. (5), we get the following update rule for the prototypes $\vec{w}_J$ and $\vec{w}_K$

$$\begin{aligned} \Delta\vec{w}_J &= +\alpha_1 \cdot \phi'(\mu(\vec{\xi})) \cdot \mu^+(\vec{\xi}) \cdot \Lambda \cdot (\vec{\xi} - \vec{w}_J) \\ \Delta\vec{w}_K &= -\alpha_1 \cdot \phi'(\mu(\vec{\xi})) \cdot \mu^-(\vec{\xi}) \cdot \Lambda \cdot (\vec{\xi} - \vec{w}_K) \end{aligned}$$

with $\mu(\vec{\xi}) = (d_J^\Lambda - d_K^\Lambda)/(d_J^\Lambda + d_K^\Lambda)$, $\mu^+(\vec{\xi}) = 2 \cdot d_K^\Lambda/(d_J^\Lambda + d_K^\Lambda)^2$, and $\mu^-(\vec{\xi}) = 2 \cdot d_J^\Lambda/(d_J^\Lambda + d_K^\Lambda)^2$
The update rule for symmetric Ω results in

$$\begin{aligned} \Delta\Omega_{lm} = &-\alpha_2 \cdot \phi'(\mu(\vec{\xi})) \cdot \\ &\Big(\mu^+(\vec{\xi}) \cdot \Big([\Omega(\vec{\xi} - \vec{w}_J)]_m(\xi_l - w_{J,l}) + [\Omega(\vec{\xi} - \vec{w}_J)]_l(\xi_m - w_{J,m})\Big) \\ &-\mu^-(\vec{\xi}) \cdot \Big([\Omega(\vec{\xi} - \vec{w}_K)]_m(\xi_l - w_{K,l}) + [\Omega(\vec{\xi} - \vec{w}_K)]_l(\xi_m - w_{K,m})\Big)\Big) \end{aligned} \tag{8}$$

which preserves the symmetry of Ω.
The update rule for non-symmetric Ω yields

$$\begin{aligned} \Delta\Omega_{lm} = &-2\,\alpha_2 \cdot \phi'(\mu(\vec{\xi})) \cdot \\ &\Big(\mu^+(\vec{\xi}) \cdot \Big((\xi_m - w_{J,m})[\Omega(\vec{\xi} - \vec{w}_J)]_l\Big) \\ &-\mu^-(\vec{\xi}) \cdot \Big((\xi_m - w_{K,m})[\Omega(\vec{\xi} - \vec{w}_K)]_l\Big)\Big) \end{aligned} \tag{9}$$

After each update, Ω is normalized to prevent the algorithm from degeneration. We set $\sum_i \Lambda_{ii} = \sum_{ij} \Omega_{ij}^2 = 1$ which fixes the sum of diagonal elements and, thus, the sum of

eigenvalues of Λ.
Depending on the dimensionality of Ω we term these learning rules Generalized Matrix LVQ($M \times N$) (GMLVQ($M \times N$)) and Generalized Matrix LVQ($N \times N$) (GMLVQ($N \times N$)), respectively.
Note, that Ω realizes a coordinate transformation to a new feature space of dimensionality $M \leq N$. The metric d^{Λ} corresponds to the Euclidean distance in this new coordinate system. This can be seen by rewriting Eq. (2) as follows:

$$d^{\Lambda}(\vec{w}, \vec{\xi}) = (\Omega(\vec{\xi} - \vec{w}))^2$$

Thus, the algorithm is not restricted to the original set of features any more to classify the data. The system is able to detect alternative directions in feature space which provide more discriminative power to separate the classes. Choosing $M < N$ implies that the classifier is restricted to a reduced number of features compared to the original input dimensionality of the data. Consequently, rank$(\Lambda) \leq M$ and at least $(N - M)$ eigenvalues of Λ are equal to zero. Since in many applications, the intrinsic dimensionality of the data is smaller than the original number of features, this approach does not necessarily constrict the performance of the classifier extensively. In addition, it can be used to derive low-dimensional representations of high-dimensional data.
Note that we can work with one full matrix Λ which accounts for a transformation of the entire input space, or alternatively, with local matrices attached to the individual prototypes. In the latter case, the squared distance of data point $\vec{\xi}$ from a prototype $\vec{w}_j$ is computed as $d^{\Lambda^j}(\vec{w}_j, \vec{\xi}) = (\vec{\xi} - \vec{w}_j)^T \Lambda^j (\vec{\xi} - \vec{w}_j)$. Localized matrices have the potential to take into account correlations which can vary between different classes or regions in feature space. We refer to this general version as Localized GMLVQ (LGMLVQ).

3 Motivation

The standard motivation for regularization is to prevent a learning system from over-fitting, i.e. the overly specific adaptation to the given training set. In previous applications of GMLVQ we observe only weak over-fitting effects. Nevertheless, restricting the adaptation of relevance matrices as outlined above can help to improve generalization ability in some cases. A more important reasoning behind the suggested regularization is the following: In previous experiments with different metric adaptation schemes in Learning Vector Quantization it has been observed, that the algorithms show a tendency to over-simplify the classifier [BBL07, SBH07a], i.e. the computation of the distance values is finally based on a strongly reduced number of features compared to the original input dimensionality of the data. In case of matrix learning, this convergence behaviour can be derived analytically for strongly simplified model situations. The elaboration of these considerations is onging work and will be topic of a forthcoming publication. Certainly, the observations desribed above indicate that the arguments are still valid under more general conditions. Frequently, there is only one feature remaining at the end of training. Depending on the adaptation of a relevance vector or a relevance matrix, this results in a single non-zero relevance factor or eigenvalue, respectively. Observing the devolution of the relevances or eigenvalues in such a situation shows that the classification error either remains constant while the metric still adapts to the data, or the over-simplification causes a degrading classification performance on training and test set. Note that these observations do not reflect over-fitting,

since training and test error increase concurrently. In case of the cost-function based algorithms this effect could be explained by the fact that a minimum of the cost function does not necessarily coincide with an optimum in matters of classification performance. Note that the term $\phi((d_J^\lambda - d_K^\lambda)/(d_J^\lambda + d_K^\lambda))$ in Eq. (1) is smaller, the larger the difference of the distance from a correct compared to an incorrect prototype. While this effect is desirable to achieve a large separation margin, it has unwanted effects when combined with metric adaptation: it causes the risk of a complete deletion of dimensions if they contribute only minor parts to the classification. This way, the classification accuracy might be severely reduced in exchange for sparse, 'over-simplified' models. But over-simplification is also observed in training with heuristic algorithms [BBL07]. Training of relevance vectors seems to be more sensitive to this effect than matrix adaptation. The determination of a new direction in feature space allows more freedom than the restriction to one of the original input features. Nevertheless, degrading classification performance can also be expected for matrix adaptation. Thus, it may be reasonable to improve the learning behavior of the GMLVQ-algorithm by preventing strong decays in the eigenvalue profile of Λ.
In addition, extreme eigenvalue settings can invoke numerical instabilities. An example scenario, which involves an artificial data set, will be presented in the Sec. 5.1. Our regularization scheme prevents the matrix Λ from becoming singular or, in the generalized case of rank limited GMLVQ, maintains a number of non-zero eigenvalues. As we will demonstrate, it thus overcomes the above mentioned instability problem.

4 Regularized Cost Function

In order to derive relevance matrices with less distinct eigenvalue profiles, we make use of the fact that maximizing the determinant of an arbitrary, quadratic matrix $A \in \mathbb{R}^{N\times N}$ with eigenvalues $\nu_1, \ldots, \nu_N$ suppresses large differences between the ν_i. Note that $\det(A) = \prod_i \nu_i$ which is maximized by $\nu_i = 1/N, \forall i$ under the constraint $\sum_i \nu_i = 1$. Hence, maximizing $\det(\Lambda)$ seems to be an appropriate strategy to manipulate the eigenvalues of Λ in GMLVQ the desired way, when Λ is non-singular. However, since $\det(\Lambda) = 0$ holds for $\Omega \in \mathbb{R}^{M\times N}$ with $M < N$, this approach cannot be applied when the computation of Λ is based on a rectangular matrix Ω. But note, that the first M eigenvalues of $\Lambda = \Omega^T\Omega$ are equal to the eigenvalues of $\Omega\Omega^T \in \mathbb{R}^{M\times M}$. Hence, maximizing $\det(\Omega\Omega^T)$ imposes a tendency of the first M eigenvalues of Λ to reach the value $1/M$. Since $\det(\Lambda) = \det(\Omega^T\Omega) = \det(\Omega\Omega^T)$ holds for $M = N$, we can use the following cost function to obtain a relevance matrix Λ with balanced eigenvalues close to $1/N$ or $1/M$ respectively:

$$\tilde{f} = f - \frac{\eta}{2} \cdot \left(\ln\left(\det\left(\Omega\Omega^T\right)\right)\right) \tag{10}$$

where f is defined in Eq. (1). The regularization parameter η adjusts the importance of the different goals covered by the two terms in $\tilde{f}$.
Since the regularization term does not include the prototype positions, the update rules for w_J and w_K do not change due to the regularization. The derivative of the regular-

ization term with respect to metric parameters yields

$$\begin{aligned}\frac{\partial \ln(\det(\Omega\Omega^T))}{\partial \Omega} &= \frac{\partial \ln(\det(\Omega\Omega^T))}{\partial \det(\Omega\Omega^T)} \frac{\partial \det(\Omega\Omega^T)}{\partial \Omega\Omega^T} \frac{\partial \Omega\Omega^T}{\partial \Omega} \\ &= 2 \cdot (\Omega^+)^T\end{aligned}$$

where Ω^+ denotes the Moore-Penrose pseudo-inverse of Ω. For the proof of this derivative we refer to [PP08]. Hence, using the modified cost function, the parameters Ω_{lm} are updated as

$$\Delta\Omega_{lm} = -\alpha_2 \cdot \frac{\partial f}{\partial \Omega_{lm}} + \alpha_2 \cdot \eta \cdot \Omega^+_{ml} \tag{11}$$

where the first term of the update rule is derived in equations (8) and (9) respectively. The idea can easily be transfered to GRLVQ: the penalty term in Eq. (10) yields $\ln(\prod_i \lambda_i)$, since the weight factors λ_i in the scaled Euclidean metric correspond to the eigenvalues of Λ in GMLVQ.

5 Experiments

In the following experiments we use different methods to initialize Ω, depending on the symmetry of the matrix. A diagonal matrix is chosen as initial state, when Ω is supposed to be symmetric. The matrix elements are initialized with uniformly distributed values in the interval $[-1, 1]$ in case of non-symmetric $\Omega \in \mathbb{R}^{N\times N}$ or $\Omega \in \mathbb{R}^{M\times N}$, followed by an adequate normalization to guarantee that the eigenvalues of Λ sum up to one. To initialize the prototypes we choose the mean values of random subsets of training samples selected from each class.
The learning rates are continuously reduced in the course of training. We implement a schedule of the form

$$\alpha_{1,2}(t) = \frac{\alpha_{1,2}}{1 + c\,(t - \tau_{1,2})} \tag{12}$$

where t counts the number training epochs and $\tau_{1,2}$ denote the starting epoch of prototype and metric adaptation. The settings $c = 10^{-4}$ and $\tau_1 = 1$ hold for all experiments.

5.1 Artificial Data

In a first illustrative experiment, the technique is applied to a two-dimensional artificial data set which constitutes a binary classification problem. The classes correspond to cigar-shaped clusters with equal prior weights. Raw data is generated according to axis-aligned Gaussians with mean $\mu_1 = [1.5, 0.0]$ for class 1 and $\mu_2 = [-1.5, 0.0]$ for class 2 data, respectively. In both classes the standard deviations are $\sigma_{11} = 0.5$ and $\sigma_{22} = 3.0$. These clusters are rotated independently by the angles $\varphi_1 = \pi/4$ and $\varphi_2 = -\pi/6$ so that the two clusters intersect. To verify the results, we perform the experiments on five different independently generated data sets. One of these data sets is visualized on Fig. 1(a).
The initial learning rates are set to $\alpha_1 = 0.05$ and $\alpha_2 = 0.005$ in all experiments. We choose $\tau_2 = 30$ in Eq. (12). Hence, the learning process starts with a phase of pure prototype training, before the metric adaptation begins. Running the GMLVQ- and LGMLVQ-algorithm on these data sets we observe that the different global and local

relevance matrices become singular already after very few sweeps through the training set. In all experiments the data has to be presented for approximately five times until the matrices reach the eigenvalue settings one and zero. In case of global matrix adaptation, the resulting classifiers always show substantially different class-wise classification accuracies. The system determines a 1-dimensional subspace in which the samples belonging to one class spread only slightly around their prototype. Due to the nature of the data set, this leads to a very poor representation of the samples belonging to the second class by the respective prototype and in consequence to a weak classification performance for this class. This issue is illustrated in Fig. 1(b) and Fig. 1(e).

The optimization of individual metrics for both prototypes allows to realize that the distances d_J to the correct prototype are lying in a small range for both classes. Concurrently, the distances d_K obtain very large values for the great majority of data points, since class 1 samples show a very large variance in the space detected for the class 2 prototype and vice versa (see Fig. 1(f), Fig. 1(g)). The only samples causing misclassifications are the data points lying in the overlapping region of the two clusters. However, since they yield very small values for both distances d_J and d_K, they cause abrupt, large parameter updates for the prototypes and the matrix elements of Ω_1 and Ω_2. This leads to instable training behavior and peaks in the learning curve as can be seen in Fig. 2. In [SBH07a] the problem is corrected manually using a heuristic approach.

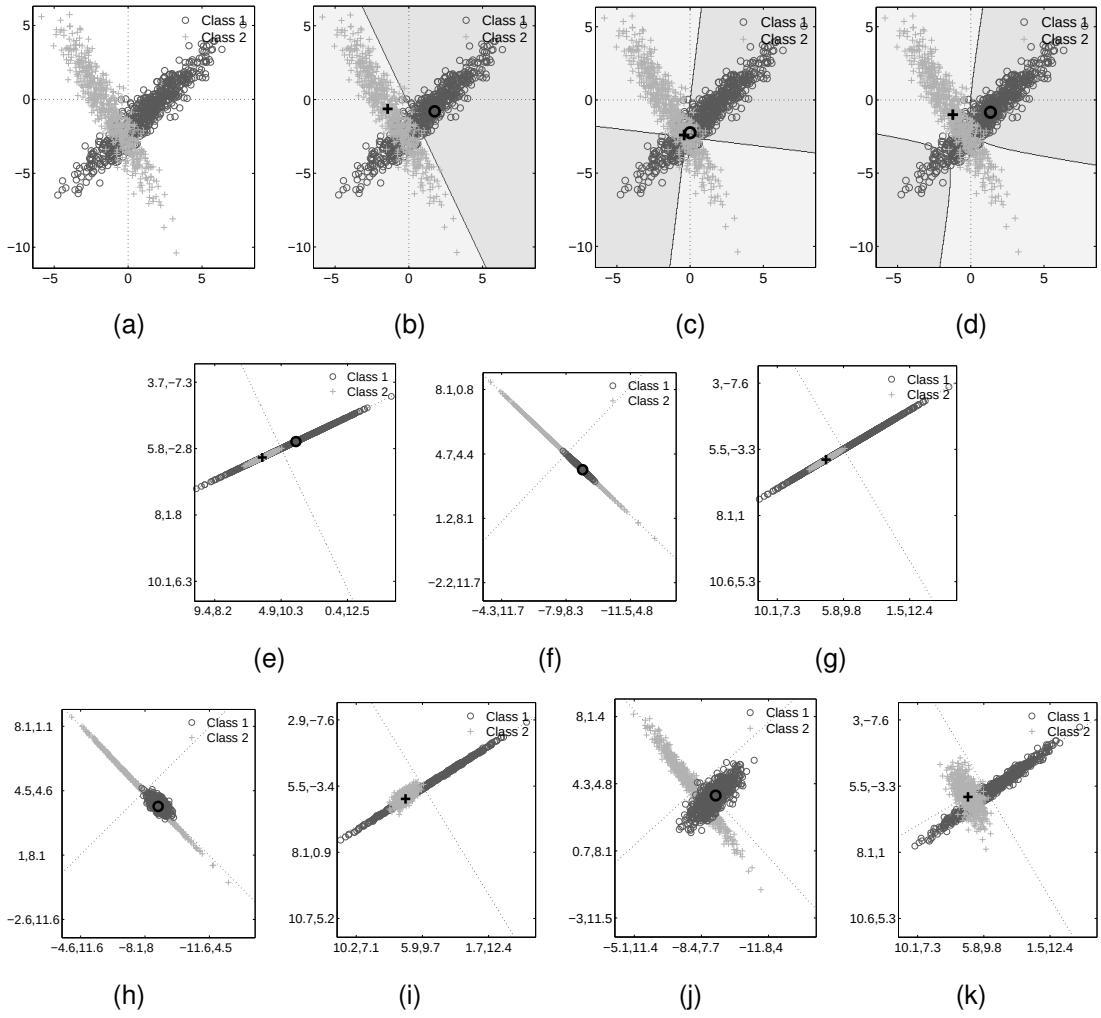
Applying the proposed regularization technique prevents the matrices $\Lambda_{1,2}$ from becoming singular and achieves a much smoother learning behavior. Choosing $\eta = 0.01$ is already sufficient to eliminate the peaks in the learning curve (see Fig. 3). The outcoming relevance matrices exhibit the eigenvalues $\text{eig}(\Lambda_{1,2}) \approx (0.99, 0.01)$. Comparing the minimum values of the error plots in Fig. 2 and Fig. 3 depicts that under these parameter settings, the regularization does not have negative impact on the classification performance.

An increasing number of misclassifications can be observed for $\eta > 0.1$. Fig. 1(d), Fig. 1(j) and Fig. 1(k) visualize the results of running LGMLVQ on the example data set with the new cost function and $\eta = 0.15$. The eigenvalue profiles of the relevance matrices obtained in these experiments are $\text{eig}(\Lambda_1) \approx (0.8, 0.2)$ and $\text{eig}(\Lambda_2) \approx (0.84, 0.16)$. The mean test error at the end of training saturates at $\varepsilon_{test} \approx 20\%$.

The problem of singular relevance matrices can also be observed when $\Lambda_{1,2}$ are derived from rectangular matrices $\Omega_{1,2}$. To construct an appropriate test case, we embed the two-dimensional data set from the previous experiment into $\mathbb{R}^5$ by adding 3 dimensions of uniformly distributed noise in $[-1, 1]$. We train individual matrices $\Omega_{1,2} \in \mathbb{R}^{2\times 5}$ which realize coordinate transformations to $\mathbb{R}^2$, since the relevant information to discriminate the classes is lying in a two-dimensional subspace. Fig. 4 depicts the learning curves for several data sets without regularization and $\eta = 0.05$. Due to the additional noise, the instabilities are not as pronounced as in the two-dimensional space. But it can also be observed that the regularization clearly reduces fluctuations and prevents numerical instabilities in the learning phase.

5.2 Diabetes Data Set

In our second experiment, we apply the algorithm to the Pima Indians Diabetes data set provided by the UCI-Repository of Machine Learning [NHBM98]. The underlying



(j)

(k)

Figure 1: (a) Artificial data set, (b) - (d) Prototypes and receptive fields, (b) GMLVQ with $\eta = 0$, (c) LGMLVQ with $\eta = 0.01$ (d) LGMLVQ with $\eta = 0.15$ (e) Training set transformed by global matrix Ω (f), (g) Training set transformed by local matrices Ω_1, Ω_2 obtained with $\eta = 0$ (h), (i) Training set transformed by local matrices Ω_1, Ω_2 obtained with $\eta = 0.01$ (j), (k) Training set transformed by local matrices Ω_1, Ω_2 obtained with $\eta = 0.15$.
In (e) - (k) the dotted lines correspond to the eigendirections of Λ_1 and Λ_2, respectively.

classification task consists of a two class problem in an 8-dimensional feature space. It has to be predicted, whether an at least 21 years old female of Pima Indian heritage shows signs of diabetes according to the World Health Organization criteria. The data set contains 768 instances, 500 class 1 samples (diabetes) and 268 class 2 samples (healthy). For our simulations we split the data set randomly into 2/3 for training and 1/3 for testing and average the results over 30 such random splits. As a preprocessing step, a z-transformation is applied to the data to normalize all features to zero mean and unit variance.
The initial learning rates are chosen as follows: $\alpha_1 = 1 \cdot 10^{-3}$, $\alpha_2 = 1 \cdot 10^{-4}$ and we set $\tau_2 = 50$ in Eq. (12). Each class is represented by one prototype. We use the weighted Euclidean metric (GRLVQ) as well as GMLVQ(8×8) with symmetric Ω, GMLVQ(8×8) with non-symmetric Ω and GMLVQ(2×8). Here the outcome of training is also a 2-

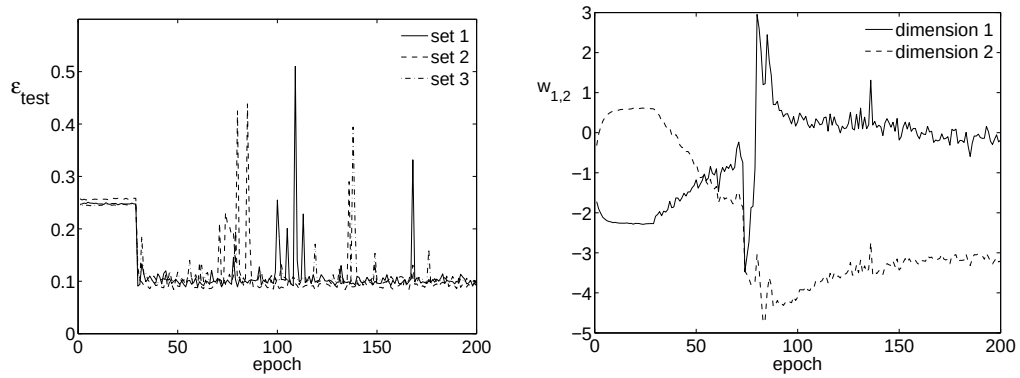


Figure 2: **Left:** Evolution of test set errors during LGMLVQ-Training on three artificial data sets with $\eta = 0$. **Right:** Coordinates of the class 2 prototype during LGMLVQ-Training on one data set with $\eta = 0$.

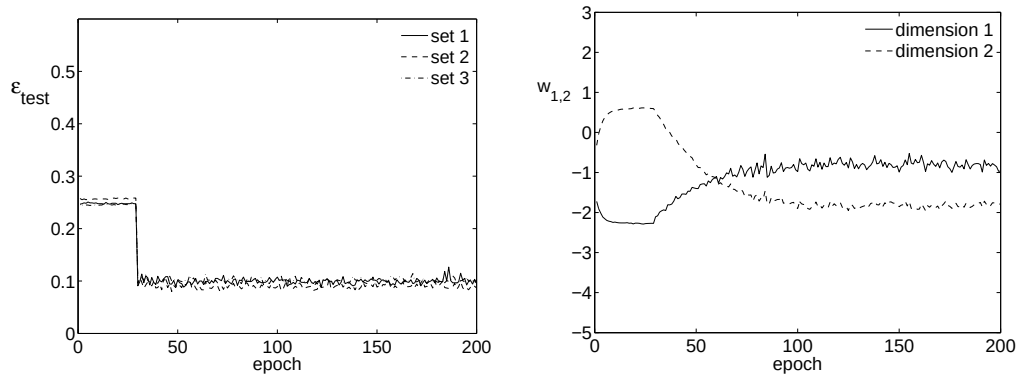


Figure 3: **Left:** Evolution of test set errors during LGMLVQ-Training on three artificial data sets with $\eta = 0.01$. **Right:** Coordinates of the class 2 prototype during LGMLVQ-Training on one data set with $\eta = 0.01$.

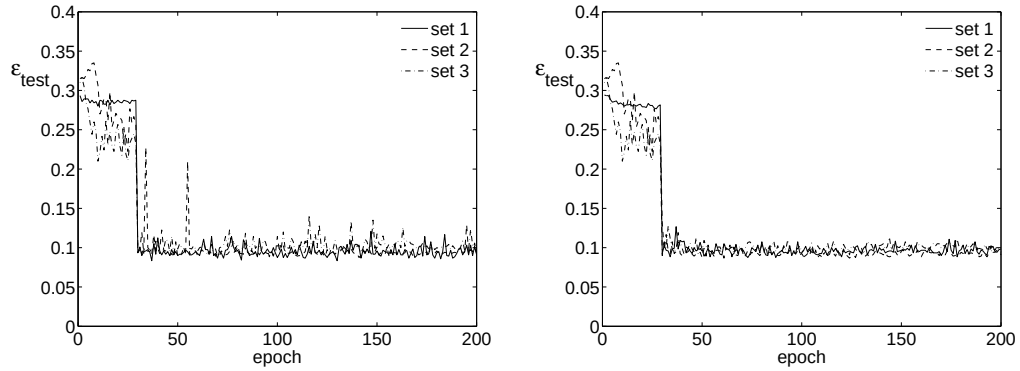


Figure 4: Evolution of test set errors during LGMLVQ-Training with $\Omega_{1,2} \in \mathbb{R}^{2\times 5}$ on three artificial data sets with three additional noise dimensions. **Left:** Training with $\eta = 0$. **Right:** Training with with $\eta = 0.05$.

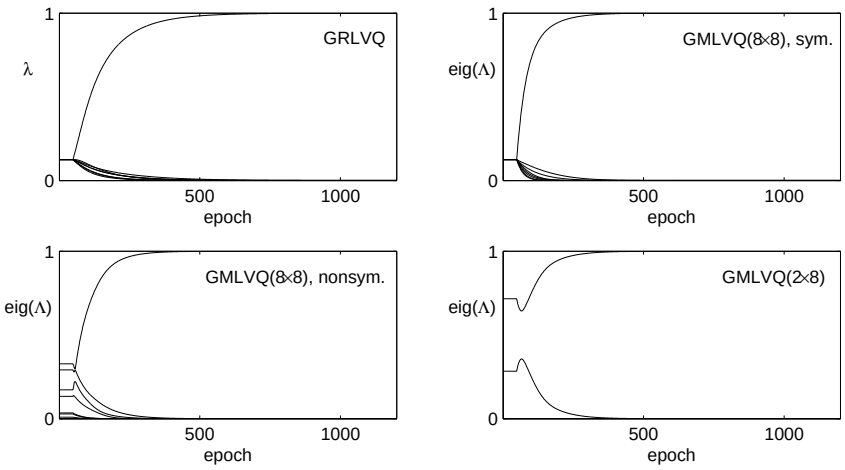


Figure 5: Diabetes data set: Evolution of the relevance values and eigenvalues during training with GRLVQ and the different GMLVQ-variants without regularization. All experiments are based on the same training data. The plots for GRLVQ, GMLVQ(8×8, sym, non-sym. reflect the weighting of 8 features. GMLVQ(2×8) is restricted to 2 features. The effect of regularization is displayed in Fig. 7.

dimensional, discriminating representation of the data. The system is trained for 1200 epochs in total.

Using the standard cost function without regularization, we observe that the metric adaptation with GRLVQ and the different GMLVQ-methods leads to an immediate selection of a single feature to classify the data. Fig. 5 visualizes examples of the evolution of relevances and eigenvalues in the course of relevance and matrix learning based on one specific training set. GRLVQ bases the classification on feature 2: Plasma glucose concentration, which is also a plausible result from the medical point of view. However, the strong feature selection results in an unstable learning behavior, as can be seen in Fig. 6, left panel. The learning curves show a distinct minimum and the error increases when training is continued. The mean test error finally saturates at $\varepsilon_{test} = 25.9\%$.

Fig. 7 (upper left panel) illustrates how the regularization parameter η influences the performance of GRLVQ. Using small values of η reduces the mean rate of misclassification on training and test sets compared to the non-regularized cost function. We observe the optimum classification performance in the training set for values around $\eta \approx 0.02$. The minimal test error $\varepsilon_{test} = 24.8\%$ is obtained with $\eta = 0.024$. However, the range of regularization parameters which achieve a comparable performance is quite small. The classifiers obtained with $\eta > 0.07$ already perform worse compared to the original GRLVQ-algorithm. Hence, the system is very sensitive with respect to to the parameter η.

In case of GMLVQ-training based on the original cost function, the strong feature selection does not result in a non-monotonic learning curve (Fig. 6, right panel). Remarkably, no significant differences in the learning behavior can be observed for the alternative settings of Ω. As depicted in Fig. 7, restricting the algorithms with the proposed regularization method can improve the test error. Note that this mild over-fitting

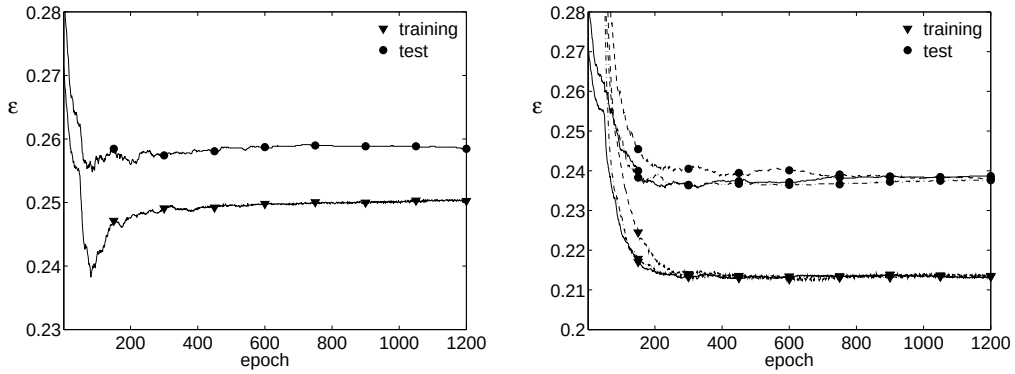


Figure 6: Diabetes data set: Evolution of averaged training and test error using the different considered algorithms without regularization. **Left:** GRLVQ-Training **Right:** GMLVQ(8×8), sym. (solid line), GMLVQ(8×8),non-sym. (dashed line), GMLVQ(2×8) (dash-dot line).

effect could not be overcome by an early stopping of the unrestricted learning procedure.

We first discuss the results obtained with the matrix adaptation schemes based on $\Omega \in \mathbb{R}^{8\times 8}$. As can be seen in Fig. 7, symmetric and non-symmetric matrices Ω show a similar response to the regularization. The two curves also share common properties with the plots obtained for GRLVQ. The classifiers performance increases for small values of η. Similar to the GRLVQ-experiments, the mean performance on the test sets reaches an optimum for $\eta \approx 0.02$. Training of symmetric matrices Ω achieves $\varepsilon_{test} = 23.4\%$ with $\eta = 0.02$ ($\varepsilon_{test} = 23.9\%$ with $\eta = 0.0$). The best performance with non-symmetric Ω constitutes $\varepsilon_{test} = 23.4\%$ obtained with $\eta = 0.025$ ($\varepsilon_{test} = 23.8\%$ for $\eta = 0$). The improvement is weaker compared to GRLVQ, but note that the parameter range of η to achieve this performance becomes wider. Furthermore, for $\eta < 0.02$, the decreasing test errors are accompanied by increasing training errors. Hence, applying the regularization technique reduces the specificity of the classifier with respect to the training data and consequently helps to prevent over-fitting.

Fig. 8 (left panel) depicts how the values of the largest relevance factor and the first eigenvalue depend on the regularization parameter. With increasing η, the values converge to $1/N$. Remarkably, the curves are very smooth. GRLVQ shows a stronger decay for small values of η and reaches the minimum $1/N$ faster compared to GMLVQ. Since the penalty term in the cost function becomes much larger for matrix adaptation with $\Omega \in \mathbb{R}^{2\times 8}$, larger values for η are necessary in order to reach the desired effect on the eigenvalues of $\Omega\Omega^T$. In our experiments, we find $\eta = 2.0$ to be necessary to achieve $\text{eig}(\Omega\Omega^T) \approx (0.5, 0.5)$ (see Fig. 8, right panel). Fig. 7 (lower right panel) shows that the error on the test set reaches a stable optimum for $\eta > 0.8$ ($\varepsilon_{test} = 23.4\%$ compared to $\varepsilon_{test} = 23.8\%$ with $\eta = 0$). The increasing test set performance is also accompanied by a decreasing performance on the training set. The plots depict, that training and test performance get closer for increasing η. Hence, the regularization supports the generalization ability of the algorithm.

As explained in Sec. 2, the coordinate transformation defined by $\Omega \in \mathbb{R}^{2\times 8}$ allows to obtain a two-dimensional representation of the data set which is particularly suitable for visualization purposes. After applying the transformation Ω to the data, the sam-

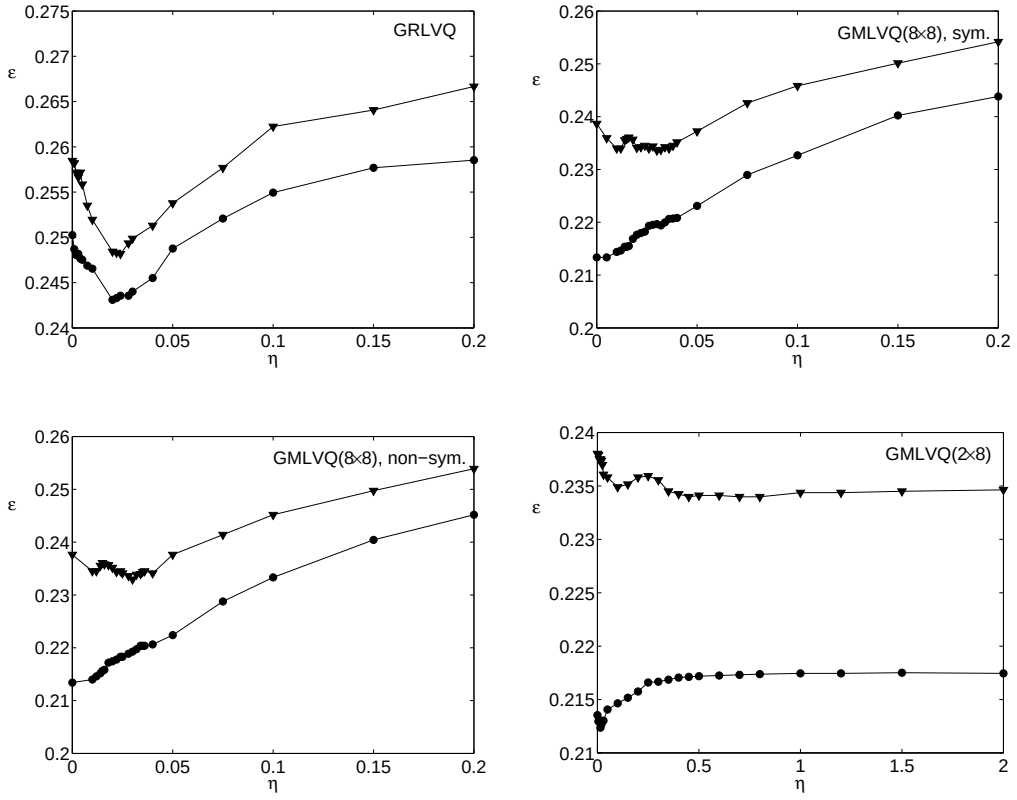


Figure 7: Diabetes data set: Mean training errors (circles) and mean test errors (triangles) after training the algorithms with different regularization parameters η.

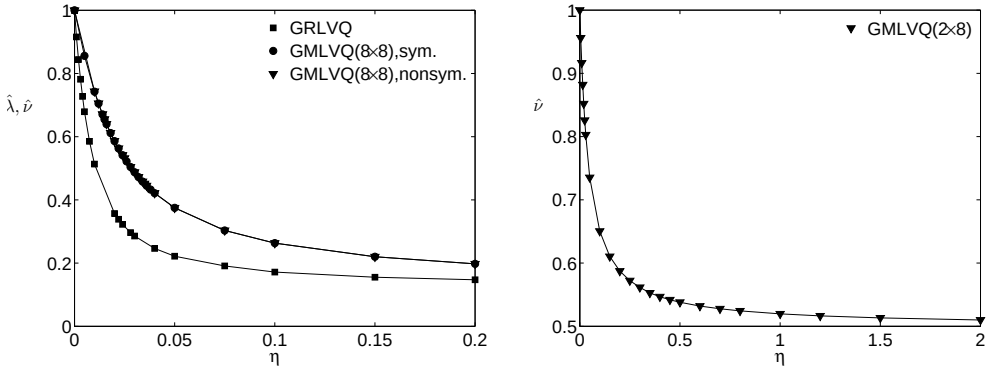


Figure 8: Diabetes data set: Dependency of the largest relevance value $\hat{\lambda}$ in GRLVQ and the largest eigenvalue $\hat{\nu}$ in GMLVQ on the regularization parameter η. The figure is based on the mean relevance factors and mean eigenvalues obtained with the different training sets after 1200 epochs. **Left:** Comparison between GRLVQ and GMLVQ(8×8) with symmetric and non-symmetric Ω. **Right:** GMLVQ(2×8).

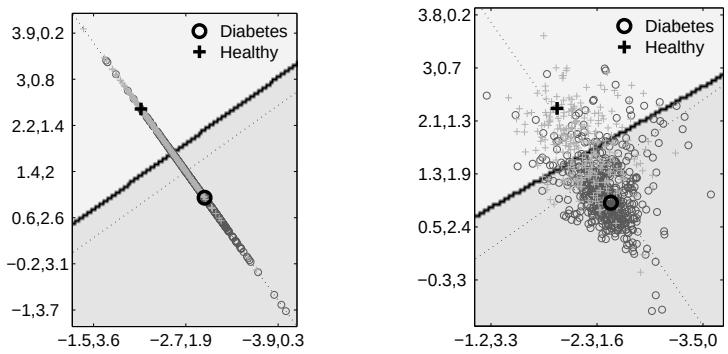


Figure 9: Diabetes data set: Two-dimensional representation of the complete data set found by GMLVQ(2×8) with $\eta = 0$ (left) and $\eta = 1.5$ (right) using one specific training set. The dotted lines correspond to the eigendirections of $\Omega\Omega^T$.

ples are scaled along the coordinate axes according to the relevances of the newly detected features, since d_Λ corresponds to the Euclidean distance in the new feature space. Due to the fact that the relevances are given by the eigenvalues of $\Omega\Omega^T$ applying the regularization technique allows to obtain visualizations which separate the classes more clearly. This property of the regularization method is illustrated in Fig. 9, which visualizes the prototypes and receptive fields which are obtained in one run. Due to the over-simplification with $\eta = 0$ the samples are projected onto a one-dimensional subspace. Visual inspection of this representation does not provide further insight into the nature of the data. On the contrary, for $\eta = 1.5$ the data is almost equally scaled in both dimensions, resulting in a discriminative visualization of the classes. In addition, we compute the error on the whole data set using these parameter settings. We observe that the performance increases in comparison to the unregularized GMLVQ(2×8). The rates of misclassification are $\varepsilon_{\eta=0} = 23.8\%$ and $\varepsilon_{\eta=1.5} = 22.4\%$.

6 Discussion

In this paper we propose a regularization scheme to improve the performance of metric adaptation techniques in Learning Vector Quantization. We focus on the adaptation of relevance vectors and relevance matrices by GRLVQ and GMLVQ, respectively. The standard GLVQ cost function is modified in order to prevent overly strong feature selection, since this effect may have negative impact on the learning behavior and classification performance. Training the prototype positions and the metric parameters is done by means of gradient descent steps with respect to the regularized cost function. The method can be applied to the original formulation of GMLVQ as well as to variants which realize a low-dimensional representation. In several experiments with artificial and real world data we observe the desired effects on the distance measure. By means of a regularization parameter it is possible to control the complexity of the relevance profile which is employed in the distance measure. We demonstrate how our regularization scheme improves the classification performance, prevents over-simplification and eliminates instabilities in the learning dynamics. Among other extensions, future projects will concern the application of the regularization method in very high-dimensional data.

There, the computational costs of the matrix inversion which is required in the relevance updates can become problematic. However, efficient techniques for the iteration of an approximate inverse can be developed which make the method also applicable to classification problems in high dimensional spaces.

References

[BBL07] BIEHL, M.; BREITLING, R.; LI, Y.: Analysis of Tiling Microarray Data by Learning Vector Quantization and Relevance Learning. In: *International Conference on Intelligent Data Engineering and Automated Learning*. Birmingham, UK : Springer LNCS, December 2007

[BGH07] BIEHL, M.; GHOSH, A.; HAMMER, B.: Dynamics and generalization ability of LVQ algorithms. In: *Journal of Machine Learning Research* 8 (2007), S. 323–360

[BHS06] BIEHL, M.; HAMMER, B.; SCHNEIDER, P.: Matrix Learning in Learning Vector Quantization / Department of Informatics, Clausthal University of Technology. 2006. – Forschungsbericht

[BHST01] BOJER, T.; HAMMER, B.; SCHUNK, D.; VON TOSCHANOWITZ, K. T.: Relevance determination in learning vector quantization. In: VERLEYSEN, M. (Hrsg.): *European Symposium on Artificial Neural Networks*, 2001, S. 271–276

[Hel02] *Bibliography on the Self-Organizing Map (SOM) and Learning Vector Quantization (LVQ)*. Neural Networks Research Centre, Helskinki University of Technology. 2002

[HSV05] HAMMER, B.; STRICKERT, M.; VILLMANN, T.: Supervised neural gas with general similarity measure. In: *Neural Processing Letters* 21 (2005), Nr. 1, S. 21–44

[HV02] HAMMER, B.; VILLMANN, T.: Generalized relevance learning vector quantization. In: *Neural Networks* 15 (2002), Nr. 8-9, S. 1059–1068

[Koh97] KOHONEN, T.: *Self-Organizing Maps*. Second. Berlin, Heidelberg : Springer, 1997

[NHBM98] NEWMAN, D. J.; HETTICH, S.; BLAKE, C. L.; MERZ, C. J. *UCI Repository of machine learning databases*. `http://archive.ics.uci.edu/ml/`. 1998

[PP08] PETERSEN, K. B.; PEDERSEN, M. S. *The Matrix Cookbook*. `http://matrixcookbook.com`. 2008

[SBH07a] SCHNEIDER, P.; BIEHL, M.; HAMMER, B.: *Adaptive relevance matrices in Learning Vector Quantization*. 2007. – Submitted

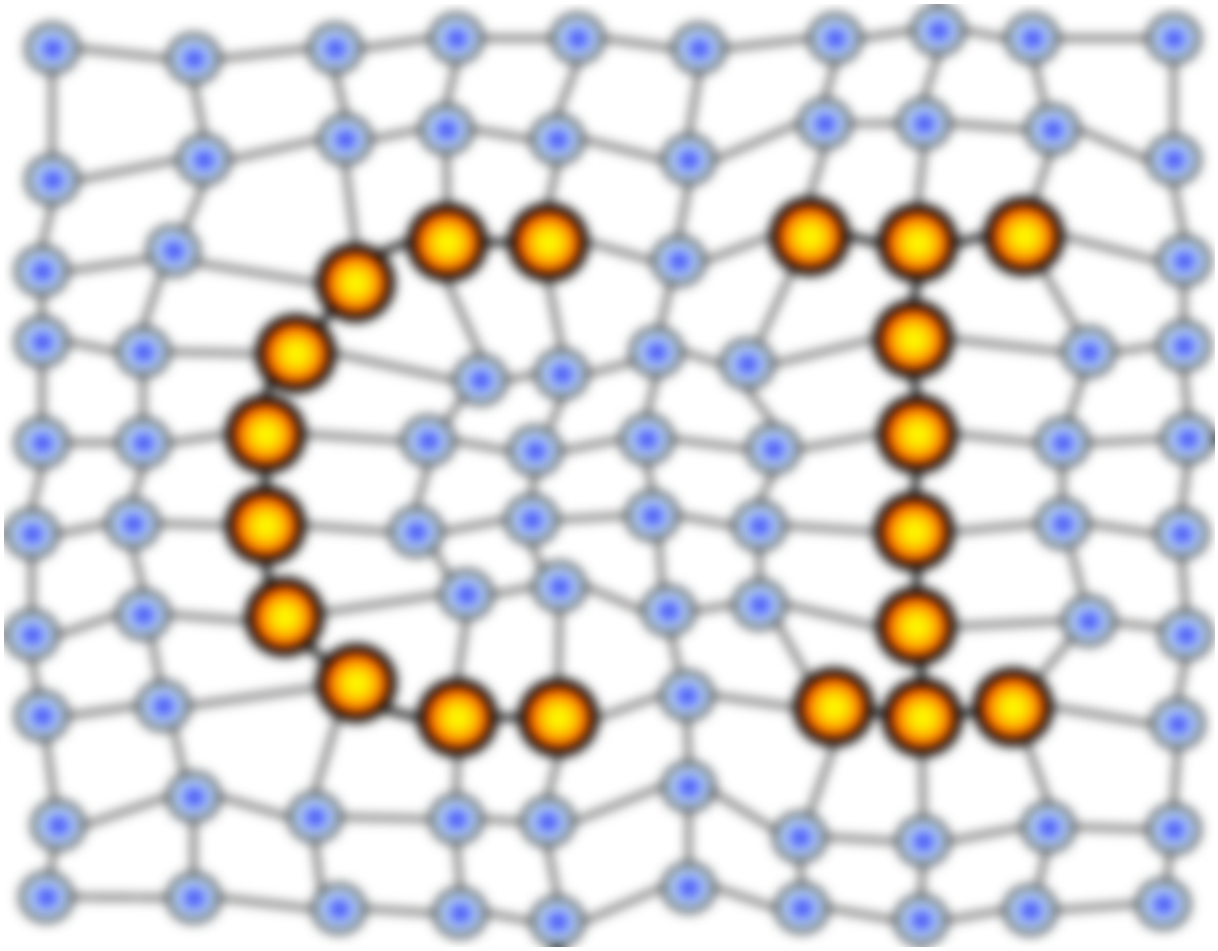
[SBH07b] SCHNEIDER, P.; BIEHL, M.; HAMMER, B.: Relevance Matrices in LVQ. In: VERLEYSEN, M. (Hrsg.): *European Symposium on Artifiacal Neural Networks*. Bruges, Belgium, April 2007, S. 37–42

[SY96] SATO, A.; YAMADA, K.: Generalized learning vector quantization. In: D. S. TOURETZKY, M. C. M. (Hrsg.); HASSELMO, M. E. (Hrsg.): *Advances in Neural Information Processing Systems 8. Proceedings of the 1995 Conference*. Cambridge, MA, USA : MIT Press, 1996, S. 423–9

[VSH06] VILLMANN, T.; SCHLEIF, F.-M.; HAMMER, B.: Comparison of Relevance Learning Vector Quantization with other Metric Adaptive Classification Methods. In: *Neural Networks* 19 (2006), S. 610–622

MACHINE LEARNING REPORTS

Report 02/2008



Impressum

Machine Learning Reports ISSN: 1865-3960

▽ Publisher/Editors

PD. Dr. rer. nat. Thomas Villmann & Dr. rer. nat. Frank-Michael Schleif
Medical Department, University of Leipzig
Semmelweisstrasse 10, D-04103 Leipzig, Germany •
http://www.uni-leipzig.de/compint

▽ Acknowledgments

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