Regularization in Matrix Relevance Learning

Petra Schneider, Kerstin Bunte, Han Stiekema, Barbara Hammer, Thomas Villmann and Michael Biehl

***Abstract*—We present a regularization technique to extend recently proposed matrix learning schemes in learning vector quantization (LVQ). These learning algorithms extend the concept of adaptive distance measures in LVQ to the use of relevance matrices. In general, metric 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. We focus on matrix learning in Generalized LVQ. Extending the cost function by an appropriate regularization term prevents the 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 benchmark classification data sets from the UCI Repository of machine learning. We demonstrate the usefulness of regularization also in the case of rank limited relevance matrices, i.e. matrix learning with an implicit, low dimensional representation of the data.**

***Index Terms*—Learning Vector Quantization, metric adaptation, cost function, regularization**

I. Introduction

Learning Vector Quantization (LVQ) as introduced by Kohonen is a particularly intuitive and simple though powerful classification scheme [1]. A set of so-called prototype vectors approximate the classes of a given data set. The prototypes parameterize a distance-based classification scheme, i.e. data is assigned to the class represented by the closest prototype. Unlike many alternative classification schemes, such as feed-forward networks or the Support Vector Machine (SVM) [2], LVQ systems are straightforward to interpret. Since the basic algorithm was introduce in 1986 [1], a huge number of modifications and extensions has been proposed, see e.g. [3], [4], [5], [6]. The methods have been used in a variety of academic and commercial applications such as image analysis, bioinformatics, medicine, etc. [7], [8].

Metric learning is a valuable technique to improve the basic LVQ approach of nearest prototype classification: a parameterized distance measure is adapted to the data to optimize the metric for the specific application. Relevance learning allows to weight the input features according to their importance for the classification task [5], [9]. Especially in case of high dimensional, heterogeneous real life data this approach turned out particularly suitable, since it accounts for irrelevant or inadequately scaled dimensions; see [10], [11] for applications. Matrix learning additionally accounts for pairwise correlations of features [6], [12]; hence, very flexible distance measures can be derived.

However, metric adaptation techniques may be subject to over-simplification of the classifier as the algorithms possibly eliminate too many dimensions. A theoretical investigation for this behavior can be found in [13].

In this work, we present a regularization scheme for metric adaptation methods in LVQ to prevent the algorithms from over-simplifying the distance measure. We demonstrate the behavior of the method by means of an artificial data set and real world applications. It is also applied in the context of rank limited relevance matrices, which realize an implicit low-dimensional representation of the data.

P. Schneider, K. Bunte, H. Stiekema and M. Biehl are with the Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, P.O. Box 407, 9700 AK Groningen, The Netherlands

B. Hammer is with the University of Bielefeld, Faculty of Technology, CITEC, 33594 Bielefeld, Germany

T. Villmann is with the Department of Mathematics/Physics/Computer Science, University of Applied Sciences Mittweida, Technikum Platz 17, 09648 Mittweida, Germany

II. Matrix learning in LVQ

LVQ aims at parameterizing a distance-based classification scheme in terms of prototypes. Assume training data $\{\xi_i, y_i\}_{i=1}^{l} \in \mathbb{R}^n \times \{1, \ldots, C\}$ are given, n denoting the data dimension 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 $\mathbf{w}_i \in \mathbb{R}^n$ and their class label $c(\mathbf{w}_i) \in \{1, \ldots, C\}$. Classification takes place by a winner takes all scheme. For this purpose, a (possibly parameterized) distance measure d is defined in $\mathbb{R}^n$. Often, the squared Euclidean metric $d(\mathbf{w}, \xi) = (\xi - \mathbf{w})^T(\xi - \mathbf{w})$ is chosen. A data point $\xi \in \mathbb{R}^n$ is mapped to the class label $c(\xi) = c(\mathbf{w}_i)$ of the prototype i for which $d(\mathbf{w}_i, \xi) \leq d(\mathbf{w}_j, \xi)$ 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.

Training of the prototype positions in feature space is often guided by heuristic update rules, e.g. in LVQ1 and LVQ2.1 [1]. Alternatively, researchers have proposed variants of LVQ which can be derived from an underlying cost function. Generalized LVQ (GLVQ) [3] e.g., is based on a heuristic cost function which can be related to a maximization of the hypothesis margin of the classifier. Mathematically well-founded alternatives were proposed in [4] and [14]: the cost functions of Soft LVQ and Robust Soft LVQ are based on a statistical modelling of the data distribution by a mixture of Gaussians, and training aims at optimizing the likelihood.

However, all these methods rely on a fixed distance, e.g. the standard Euclidean distance which may be inappropriate if the data does not display a Euclidean characteristic. The squared weighted Euclidean metric $d^{\lambda}(\mathbf{w}, \xi) = \sum_i \lambda_i (\xi_i - w_i)^2$ with $\lambda_i \geq 0$ and $\sum_i \lambda_i = 1$ allows to use prototype based learning also in the presence of high dimensional data with features of different, yet a priori unknown, relevance. Extensions of LVQ1 and GLVQ with respect to this metric were proposed

in [5], [9], called Relevance LVQ (RLVQ) and Generalized Relevance LVQ (GRLVQ).

Matrix learning in LVQ schemes was introduced in [6], [12]. Here, the Euclidean distance is generalized by a full matrix Λ of adaptive relevances. The new metric reads

$$d^{\Lambda}(\mathbf{w},\xi) = (\xi-\mathbf{w})^T \Lambda\,(\xi-\mathbf{w}), \tag{1}$$

where Λ is an $n \times n$ matrix. The above dissimilarity measure only corresponds to a meaningful distance, if Λ is positive semi-definite. This can be achieved by substituting $\Lambda = \Omega^T\Omega$, where $\Omega \in \mathbb{R}^{m\times n}$ with $m \leq n$ is an arbitrary matrix. Hence, the distance measure reads

$$d^{\Lambda}(\mathbf{w},\xi) = \sum_{i,j}^{n}\sum_{k}^{m}(\xi_i - w_i)\Omega_{ki}\Omega_{kj}(\xi_j - w_j). \tag{2}$$

Note, that Ω realizes a coordinate transformation to a new feature space of dimensionality $m \leq n$. The metric d^{Λ} corresponds to the squared Euclidean distance in this new coordinate system. This can be seen by rewriting Eq. (1) as follows:

$$d^{\Lambda}(\mathbf{w},\xi) = \left[(\xi-\mathbf{w})^T\Omega^T\right]\left[\Omega(\xi-\mathbf{w})\right].$$

Using this distance measure, the LVQ classifier 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, $\mathrm{rank}(\Lambda) \leq m$ and at least $(n-m)$ eigenvalues of Λ are equal to zero. In many applications, the intrinsic dimensionality of the data is smaller than the original number of features. Hence, 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 [15].

Moreover, it is possible to work with local matrices attached to the individual prototypes. In this case, the squared distance of data point ξ from the prototype $\mathbf{w}_j$ reads $d^{\Lambda_j}(\mathbf{w}_j,\xi) = (\xi - \mathbf{w}_j)\Lambda_j(\xi-\mathbf{w}_j)$. Localized matrices have the potential to take into account correlations which can vary between different classes or regions in feature space.

LVQ schemes which optimize a cost function can easily be extended with respect to the new distance measure. To obtain the update rules for the training algorithms, the derivatives of (1) with respect to $\mathbf{w}$ and Ω have to be computed. We obtain

$$\frac{\partial d^{\Lambda}(\mathbf{w},\xi)}{\partial \mathbf{w}} = -2\,\Lambda\,(\xi-\mathbf{w}) = -2\,\Omega^T\Omega\,(\xi-\mathbf{w}), \tag{3}$$

$$\frac{\partial d^{\Lambda}(\mathbf{w},\xi)}{\partial \Omega_{lm}} = 2\sum_i(\xi_i - w_i)\Omega_{li}(\xi_m - w_m). \tag{4}$$

Note however that Eq. (4) only holds for an unstructured matrix Ω. In the special case of quadratic, symmetric Ω, the off-diagonal elements cannot be varied independently. In consequence, diagonal and off-diagonal elements yield different derivatives. However, this special case is not considered in this study. In the following, we always refer to the most general case of arbitrary $\Omega \in \mathbb{R}^{m\times n}$.

Additionally, in the course of training, Λ has to be normalized after every update step to prevent the learning algorithm from degeneration. Possible approaches are to set $\sum_i \Lambda_{ii}$ or $\det(\Lambda)$ to a fixed value, hence, either the sum of eigenvalues or the product of eigenvalues is constant.

In this paper, we focus on matrix learning in Generalized LVQ. In the following, we shortly derive the learning rules.

A. Matrix learning in Generalized LVQ

Matrix learning in GLVQ is derived as a minimization of the cost function

$$E = \sum_{i=1}^{l}\phi\left(\frac{d_J^{\Lambda}(\xi_i) - d_K^{\Lambda}(\xi_i)}{d_J^{\Lambda}(\xi_i) + d_K^{\Lambda}(\xi_i)}\right), \tag{5}$$

where ϕ is a monotonic function, e.g. the logistic function or the identity, $d_J^{\Lambda}(\xi) = d^{\Lambda}(\mathbf{w}_J,\xi)$ is the distance of data point ξ from the closest prototype $\mathbf{w}_J$ with the same class label y, and $d_K^{\Lambda}(\xi) = d^{\Lambda}(\mathbf{w}_K,\xi)$ is the distance from the closest prototype $\mathbf{w}_K$ with any class label different from y. Taking the derivatives with respect to the prototypes and the metric parameters yields a gradient based adaptation scheme. Using Eq. (3), we get the following update rule for the prototypes $\mathbf{w}_J$ and $\mathbf{w}_K$

$$\Delta\mathbf{w}_J = +\alpha_1\cdot\phi'(\mu(\xi))\cdot\mu^+(\xi)\cdot\Lambda\cdot(\xi-\mathbf{w}_J), \tag{6}$$

$$\Delta\mathbf{w}_K = -\alpha_1\cdot\phi'(\mu(\xi))\cdot\mu^-(\xi)\cdot\Lambda\cdot(\xi-\mathbf{w}_K), \tag{7}$$

with $\mu(\xi) = (d_J^{\Lambda} - d_K^{\Lambda})/(d_J^{\Lambda} + d_K^{\Lambda})$, $\mu^+(\xi) = 4\cdot d_K^{\Lambda}/(d_J^{\Lambda} + d_K^{\Lambda})^2$, and $\mu^-(\xi) = 4\cdot d_J^{\Lambda}/(d_J^{\Lambda} + d_K^{\Lambda})^2$; α_1 is the learning rate for the prototypes. Throughout the following, we use the identity function $\phi(x) = x$ which implies $\phi'(x) = 1$. The update rule for non-structured Ω results in

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

where α_2 is the learning rate for the metric parameters. Each update is followed by a normalization step to prevent the algorithm from degeneration. We call the extension of GLVQ defined by Eq.s (6), (7) and (8) Generalized Matrix LVQ (GMLVQ) [6].

In our experiments, we also apply local matrix learning in GLVQ with individual matrices Λ_j attached to each prototype; again, the training is based on non-structured Ω_j. In this case, the learning rules for the metric parameters yield

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

$$\begin{aligned}\Delta\Omega_{K,lm} = &+\alpha_2\cdot\phi'(\mu(\xi))\cdot \\ &\mu^-(\xi)\cdot\Big((\xi_m - w_{K,m})[\Omega_K(\xi-\mathbf{w}_K)]_l\Big).\end{aligned} \tag{10}$$

Using this approach, the update rules for the prototypes also include the local matrices. We refer to this method as localized GMLVQ (LGMLVQ) [6].

III. 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 matrix learning in LVQ we observed only weak over-fitting effects. Nevertheless, restricting the adaptation of relevance matrices 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 [16], [6], 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 in LVQ1, this convergence behavior can be derived analytically under simplifying assumptions [13]. The elaboration of these considerations is ongoing work and will be topic of further forthcoming publications. Certainly, the observations described above indicate that the arguments are still valid under more general conditions. Frequently, there is only one linear combination of features 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 evolution 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 numerator in Eq. (5) is smaller than 0 iff the classification of the data point is correct. The smaller the numerator, the greater the security of classification, i.e. the difference of the distances to the closest correct and wrong 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 [16]. 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 matrix adaptation algorithms by preventing strong decays in the eigenvalue profile of Λ.

In addition, extreme eigenvalue settings can invoke numerical instabilities in case of GMLVQ. An example scenario, which involves an artificial data set, will be presented in the Sec. V-A. Our regularization scheme prevents the matrix Λ from becoming singular. As we will demonstrate, it thus overcomes the above mentioned instability problem.

IV. Regularized Cost Function

In order to derive relevance matrices with more uniform 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 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, if the computation of Λ is based on a rectangular matrix Ω. However, 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 propose the following regularization term θ in order to obtain a relevance matrix Λ with balanced eigenvalues close to $1/n$ or $1/m$ respectively:

$$\theta = \ln\left(\det\left(\Omega\Omega^T\right)\right). \tag{11}$$

The approach can easily be applied to any LVQ algorithm with an underlying cost function E. Note that θ has to be added or subtracted depending on the character of E. The derivative with respect to Ω yields

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

where Ω^+ denotes the Moore-Penrose pseudo-inverse of Ω. For the proof of this derivative we refer to [17]. Since θ only depends on the metric parameters, the update rules for the prototypes are not affected.

In case of GMLVQ, the extended cost function reads

$$\tilde{E} = E - \frac{\eta}{2} \cdot \ln\left(\det\left(\Omega\Omega^T\right)\right). \tag{12}$$

The regularization parameter η adjusts the importance of the different goals covered by $\tilde{E}$. Consequently, the update rule for the metric parameters given in Eq. (8) is extended by

$$\Delta\Omega_{\theta,lm} = +\alpha_2 \cdot \eta \cdot \Omega^+_{ml}. \tag{13}$$

The regularization parameter has to be optimized by means of a validation procedure.

The concept can easily be transfered to relevance LVQ with exclusively diagonal relevance factors [5], [9]: in this case, the regularization term reads $\theta = \ln(\prod_i \lambda_i)$, because the weight factors λ_i in the scaled Euclidean metric correspond to the eigenvalues of Λ. In the experimental section, we also examine regularization in GRLVQ.

Since θ is only defined in terms of the metric parameters, it can be expected that the number of prototypes does not have significant influence on the application of the regularization technique. This claim will be verified by means of a real life classification problem in Sec. V-B3.

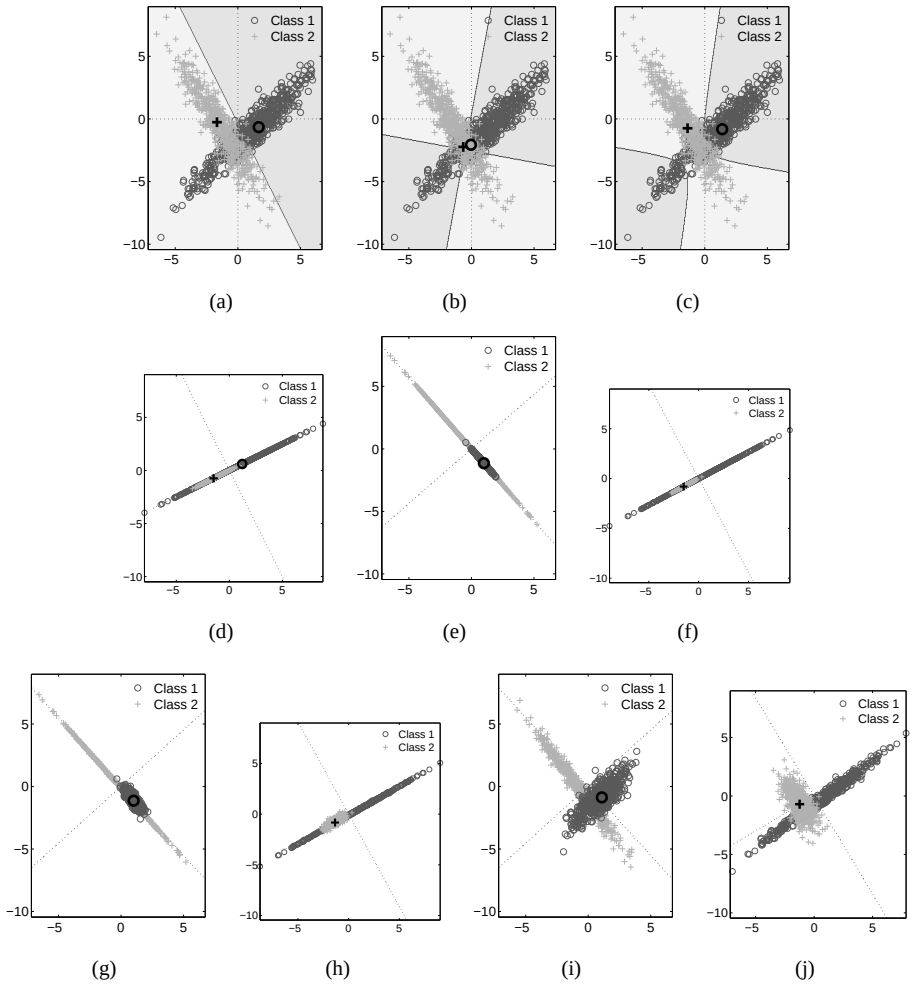

Fig. 1. *Artificial data* (a) - (c) Prototypes and receptive fields, (a) GMLVQ without regularization, (b) LGMLVQ without regularization, (c) LGMLVQ with $\eta = 0.15$, (d) Training set transformed by global matrix Ω after GMLVQ training, (e), (f) Training set transformed by local matrices Ω_1, Ω_2 after LGMLVQ training, (g), (h) Training set transformed by local matrices Ω_1, Ω_2 obtained by LGMLVQ Training with $\eta = 0.005$, (i), (j) Training set transformed by local matrices Ω_1, Ω_2 obtained by LGMLVQ Training with $\eta = 0.15$. In (d) - (j) the dotted lines correspond to the eigendirections of Λ or Λ_1 and Λ_2, respectively.

V. Experiments

In the following experiments, we always initialize the relevance matrix Λ with the identity matrix followed by a normalization step; we choose the normalization $\sum_i \Lambda_{ii} = 1$. As initial prototypes, we choose the mean values of random subsets of training samples selected from each class.

A. Artificial Data

Our first illustrative application is the artificial data set visualized in Fig. 1. It constitutes a binary classification problem in a two-dimensional space. Training and validation data are generated according to axis-aligned Gaussians of 600 samples 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 ten independently generated data sets.

At first, we focus on the adaptation of a global relevance matrix by GMLVQ. We use the learning rates $\alpha_1 = 0.01$ and $\alpha_2 = 1 \cdot 10^{-3}$ and train the system for 100 epochs. In all experiments, the behavior described in [13] is visible immediately; Λ reaches the eigenvalue settings one and zero within 10 sweeps through the training set. Hence, the system uses a one-dimensional subspace to discriminate the data. This subspace stands out due to minimal data variance around the respective prototype of one class. Accordingly, this subspace is defined by the eigenvector corresponding to the smallest eigenvalue of the class specific covariance matrix. This issue is illustrated in Fig.s 1 (a) and (d). Due to the nature of the data set, this behavior leads to a very poor representation of the samples belonging to the other class by the respective prototype which implies a very weak class-specific classification performance as depicted by the receptive fields.

However, numerical instabilities can be observed, if local relevance matrices are trained for this data set. In accordance with the theory in [13], the matrices become singular in only a small number of iterations. Projecting the samples onto the second eigenvector of the class specific covariance matrices allows to realize minimal data variance around the respective prototype for both classes (see Fig.s 1 (e), (f)). Consequently,

the great majority of data points obtains very small values d_J and comparably large values d_K. But samples lying in the overlapping region yield very small values for both distances d_J and d_K. In consequence, these data cause abrupt, large parameter updates for the prototypes and the matrix elements (see Eq.s (6), (7), (9), (10)). This leads to instable training behavior and peaks in the learning curve as can be seen in Fig. 2.

Applying the proposed regularization technique leads to a much smoother learning behavior. With $\eta = 0.005$, the matrices $\Lambda_{1,2}$ do not become singular and the peaks in the learning curve are eliminated (see Fig. 2). Misclassifications only occur in case of data lying in the overlapping region of the clusters; the system achieves $\varepsilon_{validation} = 9\%$. The relevance matrices exhibit the mean eigenvalues $\text{eig}(\Lambda_{1,2}) \approx (0.99, 0.01)$. Accordingly, the samples spread slightly in two dimensions after transformation with Ω_1 and Ω_2 (see Fig.s 1 (g), (h)). An increasing number of misclassifications can be observed for $\eta > 0.1$. Fig.s 1 (c), (i), (j) visualize the results of running LGMLVQ with the new cost function and $\eta = 0.15$. The mean eigenvalue profiles of the relevance matrices obtained in these experiments are $\text{eig}(\Lambda_1) \approx (0.83, 0.17)$ and $\text{eig}(\Lambda_2) \approx (0.84, 0.16)$. The mean test error at the end of training saturates at $\varepsilon_{validation} \approx 13\%$.

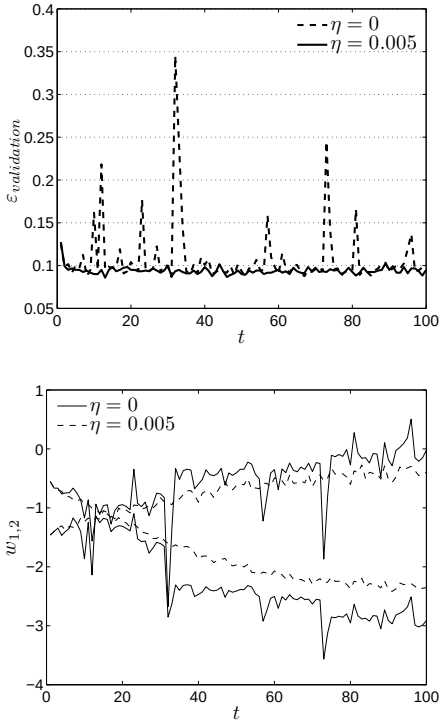

Fig. 2. *Artificial data* The plots relate to experiments on a single data set. **Top:** Evolution of error rate on validation set during LGMLVQ-Training with $\eta = 0$ and $\eta = 0.005$. **Bottom:** Coordinates of the class 2 prototype during LGMLVQ-Training with $\eta = 0$ and $\eta = 0.005$.

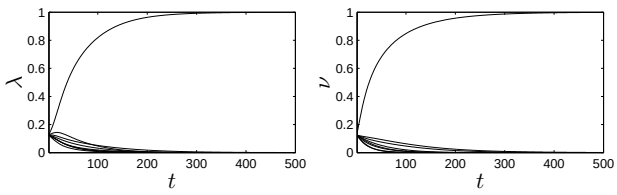

Fig. 3. *Pima indians diabetes data* Evolution of relevance values λ and eigenvalues $\nu = \text{eig}(\Lambda)$ observed during a single training run of GRLVQ (left) and GMLVQ with $\Omega \in \mathbb{R}^{8\times 8}$ (right).

B. Real Life Data

In our second set of experiments, we apply the algorithms to three benchmark data sets provided by the UCI Repository of Machine Learning [18], namely *Pima Indians Diabetes*, *Glass Identification* and *Letter Recognition*. *Pima Indians Diabetes* constitutes a binary classification problem while the latter data sets are multi-class problems.

1) Pima Indians Diabetes: The 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). As a preprocessing step, a z-transformation is applied to normalize all features to zero mean and unit variance.

We split the data set randomly into 2/3 for training and 1/3 for validation and average the results over 30 such random splits. We approximate the data by means of one prototype per class. The learning rates are chosen as follows: $\alpha_1 = 1 \cdot 10^{-3}$, $\alpha_2 = 1 \cdot 10^{-4}$. The regularization parameter is chosen from the interval [0, 1.0]. We use the weighted Euclidean metric (GRLVQ) and GMLVQ with $\Omega \in \mathbb{R}^{8\times 8}$ and $\Omega \in \mathbb{R}^{2\times 8}$. The system is trained for 500 epochs in total.

Using the standard GLVQ cost function without regularization, we observe that the metric adaptation with GRLVQ and GMLVQ leads to an immediate selection of a single feature to classify the data. Fig. 3 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.

Fig. 4 (upper 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 validation sets compared to the non-regularized cost function. We observe the optimum classification performance on the validation sets for $\eta \approx 0.03$; the mean error rate constitutes $\varepsilon_{validation} = 25.2\%$. However, the range of regularization parameters which achieve a comparable performance is quite small. The classifiers obtained with $\eta > 0.06$ already perform worse compared to the original GRLVQ-algorithm. Hence, the system is very sensitive with respect to the parameter η.

Next, we discuss the GMLVQ results obtained with $\Omega \in \mathbb{R}^{8\times 8}$. As depicted in Fig. 4 (middle), restricting the algorithm with the proposed regularization method improves the classification of the validation data slightly; the mean performance on the validation sets increases for small values of η and reaches

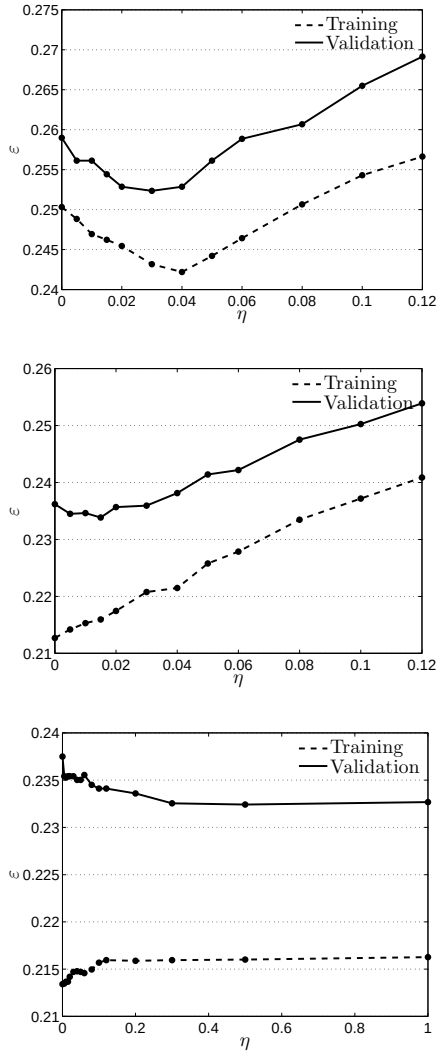

Fig. 4. *Pima indians diabetes data* Mean error rates on training and validation sets after training different algorithms with different regularization parameters η. **Top:** GRLVQ. **Middle:** GMLVQ with $\Omega \in \mathbb{R}^{8\times 8}$. **Bottom:** GMLVQ with $\Omega \in \mathbb{R}^{2\times 8}$.

$\varepsilon_{validation} \approx 23.4\%$ with $\eta = 0.015$. The improvement is weaker compared to GRLVQ, but note that the decreasing validation error is accompanied by an increasing training error. Hence, the specificity of the classifier with respect to the training data is reduced; the regularization helps to prevent over-fitting. Note that this over-fitting effect could not be overcome by an early stopping of the unrestricted learning procedure.

Similar observations can be made for GMLVQ with $\Omega \in \mathbb{R}^{2\times 8}$; the regularization slightly improves the performance on the validation data while the accuracy on the training data is degrading (see Fig. (4), bottom). 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 $\Omega\Omega^T$. The plot in Fig. (4) depicts that the mean error on the validation sets reaches a stable optimum for $\eta > 0.3$; $\varepsilon_{validation} = 23.3\%$. The increasing validation set performance is also accompanied by a decreasing performance on the training sets.

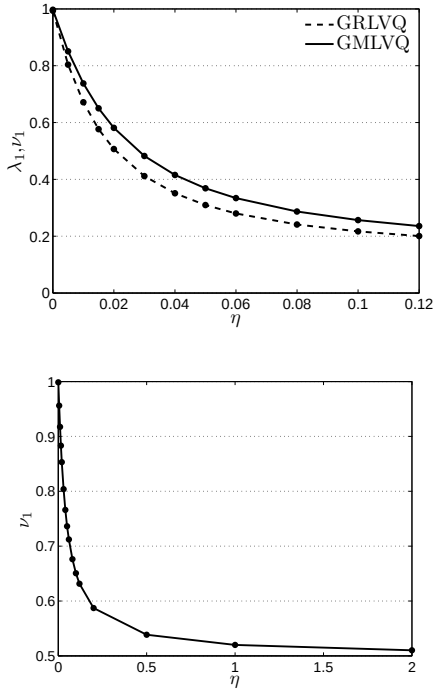

Fig. 5. *Pima indians diabetes data* Dependency of the largest relevance value λ_1 in GRLVQ and the largest eigenvalue ν_1 in GMLVQ on the regularization parameter η. The plots are based on the mean relevance factors and mean eigenvalues obtained with the different training sets at the end of training. **Top:** Comparison between GRLVQ and GMLVQ with $\Omega \in \mathbb{R}^{8\times 8}$. **Bottom:** GMLVQ with $\Omega \in \mathbb{R}^{2\times 8}$.

Fig. 5 visualizes how the values of the largest relevance factor and the first eigenvalue depend on the regularization parameter. With increasing η, the values converge to $1/N$ or $1/M$, respectively. Remarkably, the curves are very smooth.

The coordinate transformation defined by $\Omega \in \mathbb{R}^{2\times 8}$ allows to construct a two-dimensional representation of the data set which is particularly suitable for visualization purposes. In the low-dimensional space, the samples are scaled along the coordinate axes according to the features' relevances for classification. Due to the fact that the relevances are given by the eigenvalues of $\Omega\Omega^T$ the regularization technique allows to obtain visualizations which separate the classes more clearly. This effect is illustrated in Fig. 6 which visualizes the prototypes and the data after transformation with one matrix Ω obtained in a single training 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, after training with $\eta = 2.0$ the data is almost equally scaled in both dimensions, resulting in a discriminative visualization of the two classes.

SVM results reported in the literature can be found e.g. in [19], [20]. The error rates on test data vary between 19.3% and 27.2%. However, we would like to stress that our main interest in the experiments is related to the analysis of the regularization approach in comparison to original GMLVQ. For this reason, further validation procedures to optimize the

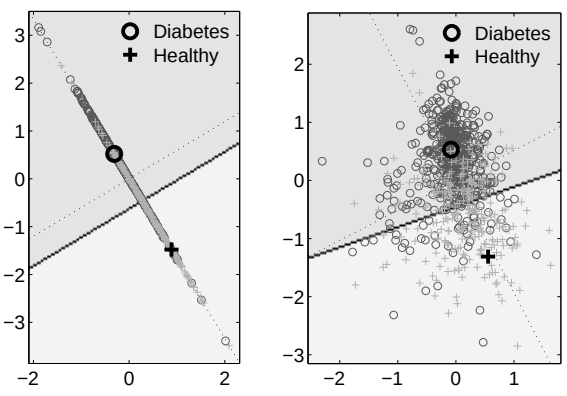

Fig. 6. *Pima indians diabetes data* Two-dimensional representation of the complete data set found by GMLVQ with $\Omega \in \mathbb{R}^{2\times 8}$ and $\eta = 0$ (left), $\eta = 2.0$ (right) obtained in one training run. The dotted lines correspond to the eigendirections of $\Omega\Omega^T$.

classifiers are not examined in this study.

2) Glass Identification: The classification task consists in the discrimination of 6 different types of glass based on 9 attributes. The data set contains 214 samples and is highly unbalanced. In case of multi-class problems, training of local matrices attached to each prototype is especially efficient. We use 80% of the data points of each class for training and the remaining data for validation. Again, a z-transformation is applied as a preprocessing step and the different classes are approximated by means of one prototype respectively. We choose the learning parameter settings $\alpha_1 = 0.01, \alpha_2 = 0.001$; the regularization parameter is selected from the interval [0, 0.4]. The following results are averaged over 200 constellations of training and validation set; we train the system in each run for 300 epochs.

On this data set, we observe that the system does not perform such a pronounced feature selection as in the previous application. The largest mean relevance after GRLVQ-training yields $\max(\lambda_i) \approx 0.3$; the largest eigenvalues after GMLVQ training constitutes $\max(\nu_i) \approx 0.5$. Nevertheless, the proposed regularization scheme is advantageous to improve the generalization ability of both algorithms as visible in Fig. (7). We observe that the mean rate of misclassification on the training data degrades for small η, while the performance on the validation data improves. This effect is especially pronounced for the adaptation of local relevance matrices. Since the data set is rather small, local GMLVQ shows a strong dependence on the actual training samples, as visible in Fig. (7), bottom. Applying the regularization reduces this effect efficiently and helps to improve the classifiers generalization ability.

Additionally, we apply GMLVQ with $\Omega \in \mathbb{R}^{2\times 9}$. We observe that the largest eigenvalue varies between 0.6 and 0.8 in different runs. The mean classification performance yields $\varepsilon_{validation} = 41\%$; the regularization does not influence the performance significantly. We observe nearly constant error rates for all tested values η. This may indicate that the intrinsic dimensionality of the data set is larger than two. Additionally, we ran the algorithm with $M = 3$ and $M = 4$. With $M = 3$ we achieve $\varepsilon_{validation} \approx 38.1\%$, $M = 4$ results in 37.2% mean

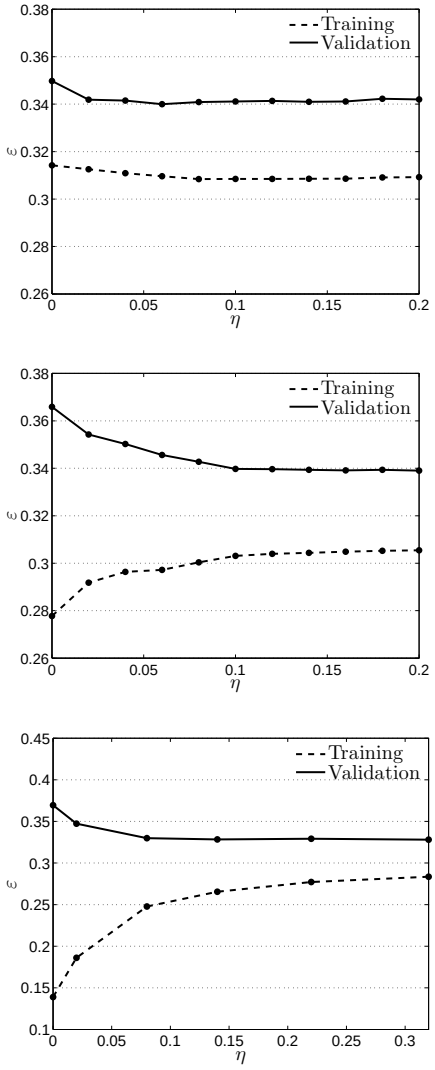

Fig. 7. *Glass identification data* Mean error rates on training and validation sets after training different algorithms with different regularization parameters η. Training of relevance matrices in GMLVQ and local GMLVQ is based on Ω, $\Omega_j \in \mathbb{R}^{9\times 9}$. **Top:** GRLVQ. **Middle:** GMLVQ. **Bottom:** Local GMLVQ.

error rate on the validation sets. Due to the regularization, the results improve sightly about 1% to 2%. Remarkably, the optimal values η already result in nearly balanced eigenvalue profile of $\Omega\Omega^T$. In this application, the best performance is achieved, if the new features are equally important for classification. The proposed regularization technique indicates such a situation.

3) Letter Recognition: The data set consists of 20 000 feature vectors encoding different attributes of black-and-white pixel displays of the 26 capital letters of the English alphabet. We split the data randomly in a training and a validation set of equal size and average our results over 10 independent random compositions of training and validation set. First, we adapt one prototype per class. We use $\alpha_1 = 1 \cdot 10^{-3}$, $\alpha_2 = 1 \cdot 10^{-4}$ and test regularization parameters from the interval [0, 0.1]. The dependence of the classification performance on the value of the regularization parameter for our GRLVQ and GMLVQ experiments are depicted in Fig. (8). It is clearly visible that

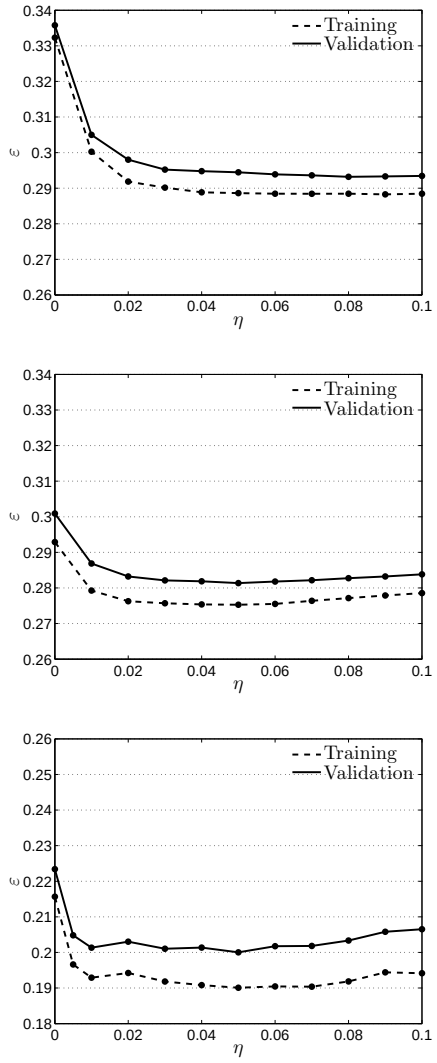

Fig. 8. *Letter recognition data set* Mean error rates on training and validation sets after training different algorithms with different regularization parameters η. **Top:** GRLVQ. **Middle:** GMLVQ with $\Omega \in \mathbb{R}^{16\times 16}$. **Bottom:** GMLVQ with $\Omega \in \mathbb{R}^{16\times 16}$ and three prototypes per class.

the regularization improves the performance for small values of η compared to the experiments with $\eta = 0$.

Compared to global GMLVQ, the adaptation of local relevance matrices improves the classification accuracy significantly; we obtain $\varepsilon_{validation} \approx 12\%$. Since no over-fitting or over-simplification effects are present in this application, the regularization does not achieve further improvements anymore.

Additionally, we perform GMLVQ training with three prototypes per class. Slightly larger learning rates $\alpha_1 = 5 \cdot 10^{-3}$ and $\alpha_2 = 5 \cdot 10^{-4}$ are used for these experiments in order to increase the speed of convergence; the system is trained for 500 epochs. Concerning the metric learning, the algorithm's behavior resembles the previous experiments with only one prototype per class. This is depicted in Fig.s 8 and 9. Already small values η effect a significant reduction of the mean rate of misclassification. Here, the optimal value η is the same for both model settings. With $\eta = 0.05$, the classification performance improves about $\approx 2\%$ compared to training with

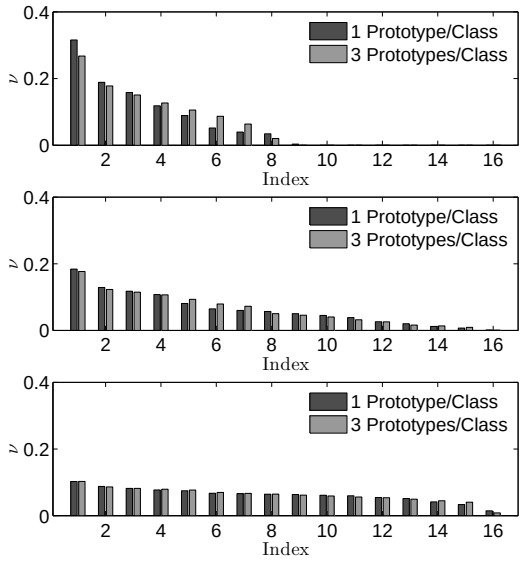

Fig. 9. *Letter recognition data set* Comparison of mean eigenvalue profiles of final matrix Λ obtained by GMLVQ training ($\Omega \in \mathbb{R}^{16\times 16}$) with different numbers of prototypes and different regularization parameters. **Top:** $\eta = 0$. **Middle:** $\eta = 0.01$. **Bottom:** $\eta = 0.05$.

$\eta = 0$. Furthermore, the shape of the eigenvalue profile of Λ is nearly independent of the codebook size (see Fig. 9). These observations support the statement that the regularization and the number of prototypes can be varied independently.

VI. Conclusion

In this article we propose a regularization technique to extend matrix learning schemes in Learning Vector Quantization. The study is motivated by the behavior analysed in [13]: matrix learning tends to perform an overly strong feature selection which may have negative impact on the classification performance and the learning dynamics. We introduce a regularization scheme which inhibits strong decays in the eigenvalue profile of the relevance matrix. The method is very flexible: it can be used in combination with any cost function and is also applicable to the adaptation of relevance vectors.

Here, we focus on matrix adaptation in Generalized LVQ. The experimental findings highlight the practicability of the proposed regularization term. It is shown in artificial and real life applications that the technique tones down the algorithm's feature selection. In consequence, the proposed regularization scheme prevents over-simplification, eliminates instabilities in the learning dynamics and improves the generalization ability of the considered metric adaptation algorithms. Beyond, our method turns out to be advantageous to derive discriminative visualizations by means of GMLVQ with a rectangular matrix Ω.

However, these effects highly depend on the choice of an appropriate regularization parameter η which has to be determined by means of a validation procedure. A further drawback constitutes the matrix inversion included in the new learning rules since it is a computationally expensive operation. Future projects will concern the application of the

regularization method on very high dimensional data. There, the computational costs of the matrix inversion can become problematic. However, efficient techniques for the iteration of an approximate pseudo-inverse can be developed which make the method also applicable for classification problems in high dimensional spaces.

REFERENCES

[1] T. Kohonen, *Self-Organizing Maps*, 2nd ed. Berlin, Heidelberg: Springer, 1997.

[2] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer, 1995.

[3] A. Sato and K. Yamada, "Generalized learning vector quantization," in *Advances in Neural Information Processing Systems 8. Proceedings of the 1995 Conference*, D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, Eds. Cambridge, MA, USA: MIT Press, 1996, pp. 423–9.

[4] S. Seo, M. Bode, and K. Obermayer, "Soft nearest prototype classification," *IEEE Transactions on Neural Networks*, vol. 14, pp. 390–398, 2003.

[5] T. Bojer, B. Hammer, D. Schunk, and K. T. von Toschanowitz, "Relevance determination in learning vector quantization," in *European Symposium on Artificial Neural Networks*, M. Verleysen, Ed., Bruges, Belgium, 2001, pp. 271–276.

[6] P. Schneider, M. Biehl, and B. Hammer, "Adaptive relevance matrices in learning vector quantization," *Neural Computation*, vol. 21, no. 12, pp. 3532–3561, 2009.

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

[8] A. Drimbarean and P. F. Whelan, "Experiments in colour texture analysis," *Pattern Recognition Letters*, vol. 22, no. 10, pp. 1161–1167, 2001.

[9] B. Hammer and T. Villmann, "Generalized relevance learning vector quantization," *Neural Networks*, vol. 15, no. 8-9, pp. 1059–1068, 2002.

[10] M. Mendenhall and E. Mereyni, "Generalized relevance learning vector quantization for classification driven feature extraction from hyperspectral data," in *in Proc. ASPRS 2006 Annual Conference and Technology Exhibition*, 2006, p. 8.

[11] T. C. Kietzmann, S. Lange, and M. Riedmiller, "Incremental grlvq: Learning relevant features for 3d object recognition," *Neurocomputing*, vol. 71, no. 13-15, pp. 2868–2879, 2008.

[12] P. Schneider, M. Biehl, and B. Hammer, "Distance learning in discriminative vector quantization," *Neural Computation*, vol. 21, no. 10, pp. 2942–2969, 2009.

[13] M. Biehl, B. Hammer, F.-M. Schleif, P. Schneider, and T. Villmann, "Stationarity of relevance matrix learning vector quantization," University of Leipzig, Tech. Rep., 2009.

[14] S. Seo and K. Obermayer, "Soft learning vector quantization," *Neural Computation*, vol. 15, no. 7, pp. 1589–1604, 2003.

[15] K. Bunte, P. Schneider, B. Hammer, F.-M. Schleif, T. Villmann, and M. Biehl, "Limited rank matrix learning and discriminative visualization," University of Leipzig, Tech. Rep. 03/2008, 2008.

[16] M. Biehl, R. Breitling, and Y. Li, "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.

[17] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," `http://matrixcookbook.com`, 2008.

[18] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz, "Uci repository of machine learning databases," `http://archive.ics.uci.edu/ml/`, 1998.

[19] C. Ong, A. A. Smola, and R. Williamson, "Learning the kernel with hyperkernels," *Journal of Machine Learning Research*, vol. 6, pp. 1043–1071, 07 2005.

[20] H. Tamura and K. Tanno, "Midpoint-validation method for support vector machine classification," *IEICE - Trans. Inf. Syst.*, vol. E91-D, no. 7, pp. 2095–2098, 2008.