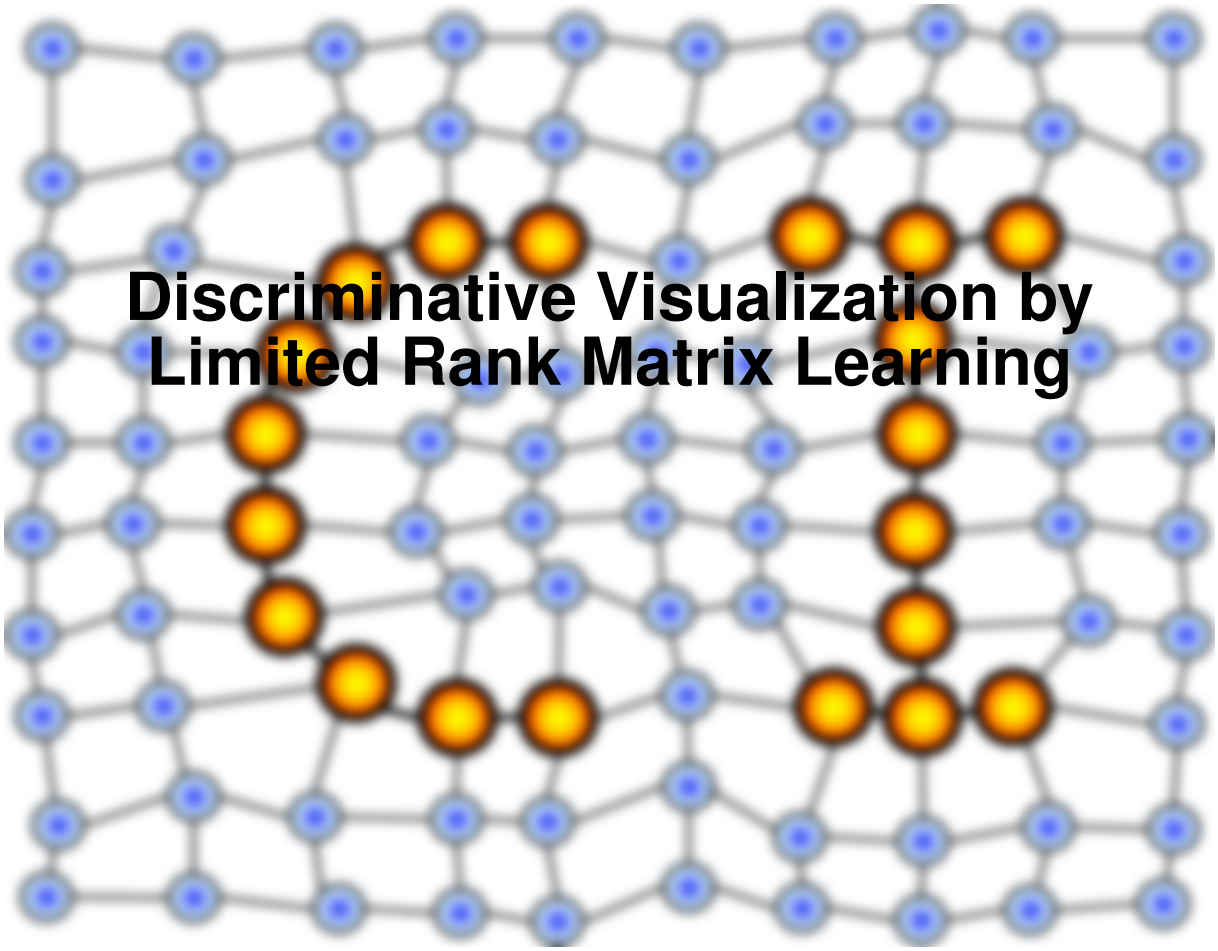
MACHINE LEARNING REPORTS



Discriminative Visualization by Limited Rank Matrix Learning

Report 03/2008
Submitted: 24.10.2008
Published: 30.10.2008

Kerstin Bunte[1] and Petra Schneider[1] and Barbara Hammer[2] and Frank-Michael Schleif[3] 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 propose an extension of the recently introduced Generalized Matrix Learning Vector Quantization (GMLVQ) algorithm. The original algorithm provides a discriminative distance measure of relevance factors, aided by adaptive square matrices, which can account for correlations between different features and their importance for the classification. We extend the scheme to matrices of limited rank corresponding to low-dimensional representations of the data. This allows to incorporate prior knowledge of the intrinsic dimension and to reduce the number of adaptive parameters efficiently. The case of two- or three-dimensional representations constitutes an efficient visualization method. The identification of a suitable projection is not treated as a preprocessing step but as an integral part of the supervised training.

1 Introduction

Learning Vector Quantization (LVQ) [Koh97] and its variants constitute a popular family of supervised, prototype-based classifiers. These algorithms have been employed successfully in a variety of scientific and commercial applications, including image analysis, bioinformatics, robotics etc. The method is easy to implement and its complexity is controlled by the user. LVQ can be applied to multiclass problems without further complication and the resulting classifiers can be interpreted intuitively. The classification of data points is based on distances to typical representatives, i.e. prototypes, which are identified in feature space.

Numerous modifications of Kohonen's original, heuristic formulation of LVQ have been suggested in the literature, aiming at better convergence properties and generalization behavior. For instance, Sato and Yamada [SY96] propose an algorithm, termed Generalized Learning Vector Quantization (GLVQ), which updates prototypes by means of gradient descent with respect to a heuristically motivated cost function. A key issue in all LVQ algorithms, with or without underlying cost function, is the choice of an appropriate similarity or distance measure. Most frequently, standard Euclidean or Minkowski metrics are employed, which are not necessarily appropriate for the given problem and data set. The fact that features can have very different meaning and magnitude in heterogeneous data, is accounted for in so-called relevance learning schemes [BHST01, HV02, HSV05] which employ adaptive scaling factors for each dimension in feature space.

In the so-called Generalized Matrix LVQ (GMLVQ)[SBH07a, SBH07b], an important extension of this concept has been introduced: a full matrix of relevances is used, which can account for correlations between different features. An adaptive self-affine transformation Ω of feature space identifies the coordinate system which is most suitable for the given classification task. The original formulation of GMLVQ employs symmetric, squared matrices. In the simplest case, one matrix is taken to define a global distance measure. Extensions to class-wise or local matrices, attached to individual prototypes, are technically straightforward and allow for the parameterization of more complex decision boundaries.

Here we propose and discuss the use of rectangular transformation matrices Ω. The corresponding relevance matrices are of bounded rank or, in other words, distances are evaluated in a space with reduced dimension. The motivation for considering this variation of GMLVQ is at least two-fold: (a) prior knowledge about the intrinsic dimension of the data can be incorporated efficiently and (b) the number of free parameters in the learning problem may be reduced significantly. In contrast with many other schemes that consider dimension reduction as a pre-processing step, our method performs the training of prototypes and the identification of a suitable transformation simultaneously. Hence, both sub-tasks are guided by the ultimate goal of implementing the desired classification scheme.

Appropriate projections into two- or three-dimensional spaces can furthermore be used for efficient visualization of labeled data. Again, it is the classification performance which directly guides the selection of the subspace. This constitutes an important difference to standard visualization strategies which implement dimensional reduction as a pre-processing step and use label information only subsequently.

Before we describe the method more formally in Sec. 3 we review GMLVQ in the following section. Next, we apply the method to a benchmark problem and study the

influence of the dimension reduction on the classification performance. In Sec. 3.4 we present example applications of our algorithm for the visualization of labeled data. We conclude by summarizing our findings and providing an outlook on perspective investigations.

2 Review of Genearlized Matrix LVQ

In this section we briefly review the Generalized Matrix LVQ algorithm [SBH07a, SBH07b]. We will assume that training is based on examples of the form $(\vec{\xi}_i, y_i) \in \mathbb{R}^N \times \{1, \ldots, C\}$, where N is the dimension of feature vectors and C is the number of classes. LVQ parameterizes the classification by means of at least C prototypes, which are chosen as typical representatives of the respective classes. They are characterized by their location in feature space $\vec{w}_i \in \mathbb{R}^N$ and the respective class label $c(\vec{w}_i) \in \{1, \ldots, C\}$. Given a parameterized distance measure $d^\Lambda(\vec{w}, \vec{\xi})$ in $\mathbb{R}^N$, the classification is done according to a "winner takes all" or "nearest prototype" scheme: Any data point $\vec{\xi} \in \mathbb{R}^N$ is assigned to the class label $c(\vec{w}_i)$ of the closest prototype i with $d^\Lambda(\vec{w}_i, \vec{\xi}) \leq d^\Lambda(\vec{w}_j, \vec{\xi})$ for all $j \neq i$. Frequently, learning corresponds to an iterative procedure which presents a single example at a time and which moves prototypes closer to (away from) data points representing the same (a different) class. In [SY96] a very flexible approach is introduced, in which the training algorithm is guided by the minimization of a cost function

$$f = \sum_i \Phi(\mu) \text{ with } \mu = \left(\frac{d_J^\Lambda - d_K^\Lambda}{d_J^\Lambda + d_K^\Lambda} \right), \tag{1}$$

where the quantities

$$\begin{aligned} d_J^\Lambda &= d^\Lambda(\vec{w}_J, \vec{\xi}_i) \quad \text{with } c(\vec{w}_J) = c(\vec{\xi}_i) \\ d_K^\Lambda &= d^\Lambda(\vec{w}_K, \vec{\xi}_i) \quad \text{with } c(\vec{w}_K) \neq c(\vec{\xi}_i) \end{aligned} \tag{2}$$

correspond to the distances of the feature vector $\vec{\xi}_i$ from the respective closest *correct (wrong)* prototype $\vec{w}_J$ ($\vec{w}_K$), respectively. In Eq. (1), Φ is a monotonic function, e.g. the logistic function or the identity $\Phi(x) = x$ which we will consider throughout the following. In GMLVQ the distance measure is specified by an $(N \times N)$ matrix, which can adapt to correlations of different features. It is of the form

$$d^\Lambda(\vec{w}, \vec{\xi}) = (\vec{\xi} - \vec{w})^T \Lambda \, (\vec{\xi} - \vec{w}) \tag{3}$$

with $\Lambda \in \mathbb{R}^{N \times N}$. The matrix Λ is assumed to be positive (semi-) definite. Hence, the measure corresponds to a (squared) Euclidean distance in an appropriately transformed space and we can substitute

$$\Lambda = \Omega^T \cdot \Omega \quad \text{with } \Omega \in \mathbb{R}^{N \times N} \quad \text{and, hence, } \; d^\Lambda(\vec{w}, \vec{\xi}) = \left[\Omega \left(\vec{\xi} - \vec{w} \right) \right]^2 \tag{4}$$

with an arbitrary matrix Ω. Specific restrictions may be imposed on Ω without loss of generality. Note that, for instance, every positive symmetric Λ has a symmetric root Ω with $\Lambda = \Omega^2$.

The original GMLVQ algorithm corresponds to a stochastic gradient descent in the cost function, Eq. (1), with respect to the prototype configuration and a symmetric matrix $\Omega \in \mathbb{R}^{N \times N}$. Gradients are evaluated with respect to the contribution of single instances $\vec{\xi}_i$ which are presented random sequentially. The algorithm has been introduced and discussed in [SBH07a, SBH07b] and will be modified in the following.

3 Limited Rank GMLVQ($M \times N$)

In the following we extend the concept of GMLVQ to the use of rectangular matrices in the distance measure. We first provide the theoretical background, than show the practical use for visualization purposes and further on we discuss the similarities and advantages in respect to related techniques.

3.1 Theoretical Background

We consider Ω to define a transformation from the original N-dimensional feature space to $\mathbb{R}^M$ with $M \leq N$ so that:

$$\Lambda = \Omega^T \Omega \quad \text{with } \Omega \in \mathbb{R}^{M \times N}. \tag{5}$$

Note that, in general, the transformation matrix Ω is not uniquely determined. The distance measure is, for instance, invariant under rotations in feature space. We identifiy a uniquely defined transformation $\hat{\Omega}$ by decomposing Λ in a canonical way: we determine the eigenvectors $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_M$ with unit length corresponding to the M (ordered) non-zero eigenvalues of Λ, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M$ and define $\hat{\Omega}$ as follows:

$$\hat{\Omega} = \left(\left[\sqrt{\lambda_1}\, \vec{v}_1, \sqrt{\lambda_2}\, \vec{v}_2, \ldots, \sqrt{\lambda_M}\, \vec{v}_M \right] \right)^\top \quad \in \mathbb{R}^{M \times N}. \tag{6}$$

In addition we choose the sign of v_i, such that the component of v_i with largest magnitude is positive.
In general, if $M > (N+1)/2$, the matrix Ω will have more independent entries then the symmetric matrix Λ. This might motivate the introduction of limitations on the number of degrees of freedom, for instance, the requirement of partial symmetry or sparsity of Ω. However, empirically we have found no evidence for overfitting or other negative effects on the learning behavior when using unrestricted transformations. Therefore we consider, throughout the following, only the general case of non-symmetric matrices Ω with $M \cdot N$ independent entries.
In order to formulate stochastic gradient descent with respect to the objective function (1) we compute the derivatives

$$\begin{aligned}
\frac{\partial d_L^\Lambda}{\partial w_{L,r}} &= -2 \cdot \sum_n^N \sum_m^M \Omega_{mr} \Omega_{mn} (\xi_n - w_{L,n}) = -2 \left[\Omega^T \Omega \right]_r (\vec{\xi} - \vec{w}_L) \\
\nabla_{\vec{w}} d_L^\Lambda &= -2 \cdot \Omega^T \Omega (\vec{\xi} - \vec{w}_L) = -2 \cdot \Lambda (\vec{\xi} - \vec{w}_L) \qquad (7) \\
\gamma^+ = \frac{\partial \mu}{\partial d_J^\Lambda} &= \frac{2 \cdot d_K^\Lambda}{(d_J^\Lambda + d_K^\Lambda)^2} \quad \text{and} \quad \gamma^- = \frac{\partial \mu}{\partial d_K^\Lambda} = \frac{-2 \cdot d_J^\Lambda}{(d_J^\Lambda + d_K^\Lambda)^2}. \qquad (8)
\end{aligned}$$

Here, $L \in \{J, K\}$ and the index J (K) refers to the closest correct (wrong) prototype $\vec{w}_J$ ($\vec{w}_K$) as introduced in Eq. (2).
For the closest correct prototype $\vec{w}_J$ and closest wrong prototype $\vec{w}_K$ one obtains an update of the form

$$\vec{w}_J^{\,\text{new}} = \vec{w}_J + \alpha_1 \cdot \Phi'(\mu(\vec{\xi})) \cdot \gamma^+ \cdot 2\Lambda(\vec{\xi} - \vec{w}_J) \tag{9}$$

$$\vec{w}_K^{\,\text{new}} = \vec{w}_K + \alpha_1 \cdot \Phi'(\mu(\vec{\xi})) \cdot \gamma^- \cdot 2\Lambda(\vec{\xi} - \vec{w}_K) \tag{10}$$

The corresponding matrix update reads

$$\frac{\partial d_L^\Lambda}{\partial \Omega_{mn}} = 2\sum_i^N (\xi_n - w_{L,n})\Omega_{mi}(\xi_i - w_{L,i}) = 2 \cdot \left[\Omega(\vec{\xi} - \vec{w}_L)\right]_m \cdot (\xi_n - w_{L,n}) \tag{11}$$

$$\begin{aligned}
\Delta\Omega_{mn} &= -\alpha_2 \cdot \Phi'(\mu(\vec{\xi})) \cdot \frac{\partial \mu}{\partial \Omega_{mn}} \\
&= -\alpha_2 \cdot \Phi' \cdot \frac{\left(\frac{\partial d_J}{\partial \Omega_{mn}} - \frac{\partial d_K}{\partial \Omega_{mn}}\right)(d_J + d_K) - \left(\frac{\partial d_J}{\partial \Omega_{mn}} + \frac{\partial d_K}{\partial \Omega_{mn}}\right)(d_J - d_K)}{(d_J + d_K)^2} \\
&= -\alpha_2 \cdot \Phi'(\mu(\vec{\xi})) \cdot \left(\gamma^+ \cdot \frac{\partial d_J^\Lambda}{\partial \Omega_{mn}} + \gamma^- \cdot \frac{\partial d_K^\Lambda}{\partial \Omega_{mn}}\right)
\end{aligned} \tag{12}$$

After each update step, the transformation matrix Ω is normalized such that

$$\sum_i \Lambda_{ii} = \sum_{mn} \Omega_{mn}^2 = 1 \tag{13}$$

and the sum of eigenvalues becomes 1.
Instead of using one global matricies Ω and Λ the formulations are easily adapted for classwise or prototypewise matricies Λ^L. The distance measure denoted by equation (3) changes for e. g. prototypewise matricies to

$$d^{\Lambda^L}(\vec{w}_L, \vec{\xi}) = (\vec{\xi} - \vec{w}_L)^T \Lambda^L (\vec{\xi} - \vec{w}_L). \tag{14}$$

For the closest correct prototype $\vec{w}_J$ and closest wrong prototype $\vec{w}_K$ one obtains an update of the form

$$\vec{w}_J^{\text{new}} = \vec{w}_J + \alpha_1 \cdot \Phi'(\mu(\vec{\xi})) \cdot \gamma^+ \cdot 2\Lambda^J(\vec{\xi} - \vec{w}_J) \tag{15}$$

$$\vec{w}_K^{\text{new}} = \vec{w}_K + \alpha_1 \cdot \Phi'(\mu(\vec{\xi})) \cdot \gamma^- \cdot 2\Lambda^K(\vec{\xi} - \vec{w}_K) \tag{16}$$

The corresponding matrix update reads

$$\Delta\Omega_{mn}^J = -\alpha_2 \cdot \Phi'(\mu(\vec{\xi})) \cdot \gamma^+ \cdot 2 \cdot \left[\Omega^J(\vec{\xi} - \vec{w}_J)\right]_m \cdot (\xi_n - w_{J,n}) \tag{17}$$

$$\Delta\Omega_{mn}^K = -\alpha_2 \cdot \Phi'(\mu(\vec{\xi})) \cdot \gamma^- \cdot 2 \cdot \left[\Omega^K(\vec{\xi} - \vec{w}_K)\right]_m \cdot (\xi_n - w_{K,n}) \tag{18}$$

Note that the learning rates α_1 and α_2 can be chosen independently. In general, we set $\alpha_1 \gg \alpha_2$ which implies that changes of the metric occur on a much slower time scale than those of the prototypes. We do not want to change the map faster, than we are running on it. This setting has proven advantageous in many implementations of relevance learning [BHST01, HV02, SBH07a].
In all practical examples considered in the following, we apply a learning rate schedule of the form

$$\alpha_1(t) = \frac{\alpha_1^{\text{start}}}{1 + (t-1)\Delta\alpha_1} \text{ and } \alpha_2(t) = \frac{\alpha_2^{\text{start}}}{1 + (t - t_M)\Delta\alpha_2}. \tag{19}$$

Here, t corresponds to the current epoch, i.e. sweep through the data set, $\alpha_{1,2}^{\text{start}}$ denotes the initial learning rates and $\Delta\alpha_{1,2}$ the strength of the annealing. Non-zero relevance

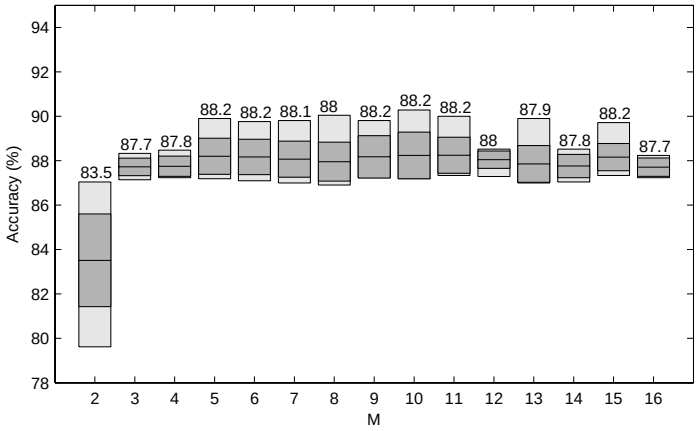


Figure 1: Performance of the GMLVQ($M\times N$) with one prototype per class as a function of M for the UCI image segmentation data set. We display the test accuracy on average over 10 random initializations, also given as a numerical value. The light shading corresponds to the interval from worst to best accuracy, the darker area marks the standard deviations.

updates are performed only after the first t_M epochs in which only the prototype positions are updated. Initial positions $\vec{w}_i(t=0)$ of the prototypes are determined by randomly selecting $1/3$ of the available feature vectors in class $c(\vec{w}_i)$ and taking the respective mean. Hence, prototypes are initially close to the class-conditional means in the training data, but with small deviations due to the random sampling. Relevance initialization is done by generating independent uniform random numbers $\Omega_{ij} \in [-1,1]$ and subsequent normalization according to Eq. (13).

3.2 Experiments

In this section we study the performance of the GMLVQ($M\times N$) algorithm on the image segmentation data set as provided in the UCI repository [NHBM98].
There, 19-dimensional feature vectors have been constructed from regions of 3×3 pixels, randomly drawn from a set of 7 manually segmented outdoor images. The features encode various attributes of the example patches, which have to be assigned to one of the following 7 classes: brickface, sky, foliage, cement, window, path, and grass. The provided data set consists of 210 feature vectors for training, with 30 instances per class. The test set comprises 300 instances per class, i.e. 2100 samples in total. We refer the reader to [NHBM98] for the details. In the data as provided by the UCI repository, features 3, 4 and 5 (region-pixel-count, short-line-density-5 and short-line-density-2) display zero variance. Hence, we omit these features and consider only the remaining 16 features. After a z-transformation, each feature displays zero mean and unit variance in the data set.
We apply in the following the GMLVQ($M\times N$) algorithm with global matrix Λ and parameters $\alpha_1^{start}=0.01$, $\Delta\alpha_1=0.0001$, $\alpha_2^{start}=0.001$, $\Delta\alpha_2=0.0001$ in the schedule (19), matrix adaptation begins in epoch $t_M=100$. Similar settings have proven successful in previous applications of the original GMLVQ algorithm to the data set [SBH07a, SBH07b].
We first study the simplest GMLVQ classifier with only one prototype per class. For sev-

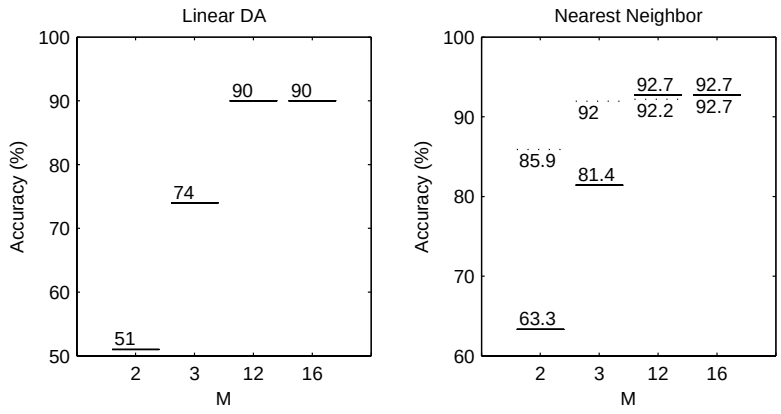


Figure 2: Performance of combinations of dimension reduction by PCA and subsequent supervised training for various M, employed to the UCI image segmentation data set. For $M = 16$, the feature vectors are used without performing a PCA. The results are to be compared with those for GMLVQ($M \times N$), see Fig. 1. Left panel: test accuracy obtained by LDA as described in the text. Right panel: test accuracies for the NN classifier using the PCA-based transformation to M dimensions (solid lines). In addition, the results after transforming the data with Ω as obtained in GMLVQ($M \times N$), the dotted lines mark the average over 10 random initialization as in Fig. 1.

eral values of M, we perform GMLVQ($M \times N$) on the given training set of 210 samples and observe the evolution of training and test accuracies within the number of 1000 epochs. We present averages and standard deviations with respect to 10 different random initializations of the prototypes and matrix Ω, as an indication of the robustness and convergence properties.
Fig. 1 shows that the GMLVQ($M \times N$) with large enough M already yields the same performance as the unrestricted variant. Only for small M we observe a clear dependence of the test accuracy on the rank of Ω, while all $M \geq 5$ display essentially the same performance. In the extreme case $M = 2$ we observe a significant drop of the generalization ability due to the serious restriction to only two non-zero eigenvalues of Λ. At the same time, the outcome of training displays a large variability: random initializations of Ω can lead to the selection of very different transformation matrices as reflected in the increased standard deviation.

3.3 Comparison with Other Methods

Here we compare the GMLVQ($M \times N$) scheme with frequently used standard procedures of comparable complexity. GMLVQ with only one prototype per class appears to be similar in spirit to the well known Linear Discriminant Analysis (LDA) [DHS00, Fri89, BC96]. In this method, a Multivariate Normal density (MVN) is fitted to the observed data in each class, here we consider a pooled estimate of the covariance matrix. Given the density estimates, the best linear decision boundaries are constructed in order to approximate Bayes optimal classification [DHS00]. The well known Nearest-Neighbor (NN) classifier serves as a second reference: Based on the standard Euclidean distance measure, any feature vector is simply assigned to the class of the closest labeled example [DHS00]. For the given data set, the extension to

K-Nearest-Neighbor schemes displays only a weak dependence on K and results will not be presented here.
The most common strategy for dimension reduction is Principal Component Analysis (PCA). In order to compare with GMLVQ($M \times N$), we apply PCA to the entire data set and obtain a low-dimensional representation in terms of the first M principal components. The projected training data is then used in LDA or serves as the reference set of the NN classifier. In the case $M = 16$, the full data set is employed without performing a PCA.
In Fig. 2, the achieved test accuracies are displayed for several values of M. For large enough dimension M, the principal components capture all relevant information and the performance of, both, LDA and NN is comparable to that of the GMLVQ($M \times N$) prescription. This finding is consistent with the M-dependence discussed in the previous paragraph.
Significant differences can be observed for small M: The dimension reduction by PCA (or any other unsupervised technique) does not take into account label information and may focus on features with large variation but little relevance for the classification. Therefore, the subsequent supervised training does not reach the quality of the GMLVQ($M \times N$) scheme with one prototype per class. Here, the complexity of the system is similar but the identification of a suitable low-dimensional representation is directly guided by the classification, which facilitates superior performance. This is easily demonstrated by replacing the PCA based transformation by the matrix Ω obtained in GMLVQ($M \times N$) see Eq. (4). Now, the simple NN system performs significantly better, as displayed in the left panel of Fig. 2. The idea of determining a discriminative transformation directly within the KNN classification scheme has been put forward in [WBS06], there without considering dimensional reduction.
GMLVQ($M \times N$) with several prototypes per class and a global relevance matrix can implement piecewise linear decision boundaries, which can exceed the complexity of LDA or similar methods significantly. In this report we do not go into detail, but as expected, the improvement of the accuracies is praticularly pronounced for small M. Even the unrestricted matrix desplays only three non-zero eigenvalues. The increased complexity due to the larer number of prootypes facilitates good performance in spite of a very simple implicit representation of the data. The use of more eigendirection could be enforced by means of a matrix regularization scheme suggested in [SBS+08b].

3.4 Visialization

The GMLVQ($M \times N$) prescription with $M = 2$ or $M = 3$ can be readily employed as a tool for the visualization of labeled data sets. In contrast to many standard methods, the tasks of identifying an appropriate subspace and implementing the actual classification is addressed in a single training phase.
The above discussed UCI segmentation data may serve as a first illustrative example. From the 10 independent runs performed with $M = 2$ to obtain the results displayed in Fig. 1 (single prototype) and for several prototypes per class, we have selected the runs that achieved the best training accuracy in order to achieve the most discriminative visualization. As mentioned above, the actual outcome can depend on the random initialization of the GMVLQ system. With a single prototype per class, a maximum accuracy of 87.4% on the entire data set is achieved. The use of 2 prototypes per class (3 in class 5) yields a best accuracy of 90.4% on the entire set.

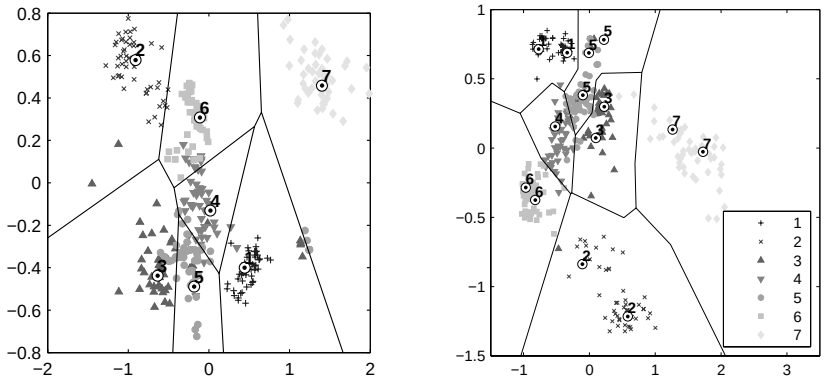


Figure 3: Visualization of the UCI segmentation data set as transformed in GLMVQ($2\times N$). The x and y axes correspond to the components of the projection $\hat{\Omega}\vec{\xi}$. For the sake of clarity we display only 50 examples per class. The left panel shows the result after training with only one prototype per class, achieving an accuracy of 87.4%. The right panel corresponds to the case of two prototypes per class (three in class 5) with 90.4% of the data classified correctly. Here we have removed one unused prototype (class 4) after training, which does not influence the classification.

Fig. 3 displays the data after the corresponding transformations. This multi-class problem allows for very good classification performance already in two dimensions. The use of several prototypes even enhances the accuracy by realizing more complex piecewise linear decision boundaries. We would like to point out once more that the identification of an appropriate projection is not treated as a pre-processing step but as an integral part of the supervised learning process. Further examples will be presented in the following in order to illustrate the concept.

Discriminative visualization can be particularly useful in the context of medical data. Here we apply the GMLVQ($M \times N$) algorithm to two gene expression data sets which were recently analysed by Faith, Mintram, and Angelova in [FMA06].

The first set concerns *small round blue cell childhood tumors*, and we refer to it as SRBCT. It comprises cDNA microarray expression levels of 50 pre-selected genes in 83 different samples [KWR+01]. The target classification assigns every sample to one of 4 tumor types.

We will refer to the second data set as NCI. It contains gene expression data from 60 cell lines from the National Cancer Institute anticancer drug screen [SRW+00]. Again 50 genes have been pre-selected and samples are to be assigned to one of 8 different types of tissue.

For details of the data sets we refer to [FMA06] and references therein. The authors present a method termed Targeted Projection Pursuit (TPP) and compare it with several existing techniques, including Multi-dimensional Scaling (MDS) [EC01], VizStruct [ZZR04], a dendrogram based method [ESBB98], and Projection Pursuit [LCKL05]. TPP is demonstrated to outperform most of these methods or to achieve at least comparable performance on the above data sets. The employed data sets as well as source codes of TPP implementations are publically available [FMA06].

First, we apply GMLVQ($2\times N$) with one prototype per class to the SRBCT data set. Results presented her are obtained after 1000 epochs with respect to the entire data set of 83 samples. We observe almost no variability with respect to random initializations

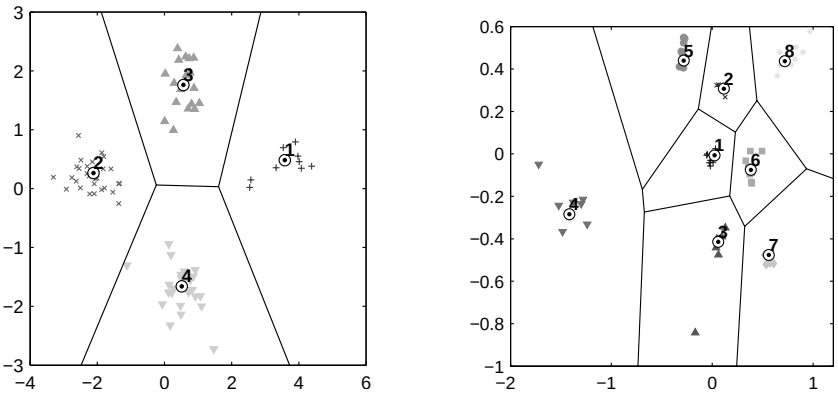


Figure 4: Two-dimensional, error-free visualizations of the SRBCT data set (left panel) and the NCI data (right panel) obtained by GMLVQ($2 \times N$) with 1 prototype per class, see the text for details.

of the system. A typical outcome is displayed in Fig. 4, left panel, the obtained $2D$ visualization perfectly separates the four classes. Error free visualizations were also obtained by Faith et *al.*, see [FMA06] for comparison.
The analogous application of GMLVQ($2 \times N$) to the NCI 8-class-problem shows slightly larger variability of results. In 10 runs with different random initialization we obtain after 1000 epochs accuracies in the range from 95.1%-100%, with an average of 97.7%. Fig. 4 (right panel) displays a perfectly separating visualization. In [FMA06], error free visualizations of the NCI data are obtained by means of TPP in combination with PCA, Projection Pursuit and subsequent LDA or KNN classification. For a visual inspection of the achieved separation we refer to Figs. 9 and 11 in [FMA06], which display either slightly overlapping classes or only very small gaps between some of them. Other methods considered in [FMA06] yield less favorable results on this data set. Most of all, we would like to point out that our method appears very simple and intuitive compared to many other suggested approaches. However, it yields comparable or even superior results.

4 Conclusions

In this report we present the GMLVQ($M \times N$) algorithm, a modification of Generalized Matrix LVQ [SBH07a]. It employs rectangular transformation matrices to implicitly represent N-dim. feature vectors in an M-dim. space. This makes it possible to limit the rank of the relevance matrices used in GMVLQ to parameterize an adaptive distance measure. The aim can be to incorporate prior knowledge of the intrinsic dimension or to reduce the number of free parameters while maintaining good classification performance. First we illustrate the approach in terms of a multi-class benchmark data set and compare with other methods of similar complexity. We demonstrate that GMVLQ($M \times N$) is an efficient method for determining discriminative, low-dimensional representations of labeled data and facilitates good generalization behavior. In GMLVQ($M \times N$), the search for the appropriate subspace is guided directly by the classification performance in a single supervised training phase. This is in contrast with classical combinations of unsupervised dimension reduction and subsequent

supervised learning.
A particular attractive application of the concept concerns the visualization of labeled data sets. Setting $M = 2$ or 3 in GMLVQ($M \times N$) provides us with a discriminative visualization of the original data set. Again, the advantage over many other methods is that the search for the suitable representation is directly integrated into the supervised training procedure. We demonstrate the usefulness of this concept in the context of several real world multi-class problems.
In this paper we have not emphasized one particularly attractive feature of relevance learning: The resulting transformation and relevance matrices can be readily interpreted and carry important information about the structure of the data. For instance, in the visualization of gene expression data, Sec. 3.4, we note that several features (intensities) essentially do not contribute to the highly discriminative linear combinations defined by $\hat{\Omega}$. This type of information provides valid insights to the application expert and should be exploited systematically.
In forthcoming projects we will also investigate several extensions of the method. So far we only limit the maximum rank of relevance matrices by choice of the parameter M, the effective dimension of the transformation can become even smaller. In applications, including visualization, it can be desirable to fix the rank and to make the system exhaust the bound. This could be done in terms of an efficient regularization method which we developed recently [SBS+08a]. Furthermore it is also possible to use local or class-wise transformation matrices. This allows for much more complex, e.g. piecewise quadratic, decision boundaries. In this frame one could then allow for different values of M in different classes or regions of feature space. The resulting scheme should be much more flexible in adapting to the structure of the data set [1].

References

[BC96] BENSMAIL, H.; CELEUX, G.: Regularized gaussian discriminant analysis through eigenvalue decomposition. In: *Journal of the American Statistical Association* 91 (1996), S. 1743–1748

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

[DHS00] DUDA, R. O.; HART, P. E.; STORK, D. G.: *Pattern Classification*. John Wiley & Sons, 2000

[EC01] EWING, R. M.; CHERRY, J. M.: Visualization of expression clusters using Sammon's non-linear mapping. In: *Bioinformatics* 17 (2001), S. 658–659

[1] Another important step is the use of local or class-wise transformation matrices. This allows for much more complex, e.g. piecewise quadratic, decision boundaries. In this frame one could then allow for different values of M in different classes or regions of feature space. The resulting scheme would be much more flexible in adapting to the structure of the data set.

Most importantly, we plan to apply GMLVQ($M \times N$) in various application domains, including the ones discussed above. In particular, efficient visualization techniques are very much needed in many different fields and disciplines.

[ESBB98] EISEN, M. B.; SPELLMAN, P. T.; BROWN, P. O.; BOTSTEIN, D.: Cluster analysis and display of genome-wide expression patterns. In: *PNAS 95* 25 (1998), S. 14863–14868

[FMA06] FAITH, J.; MINTRAM, R.; ANGELOVA, M.: Targeted Projection Pursuit for Visualising Gene Expression Data Classifications. In: *Bioinformatics* 22 (2006), S. 2667–2673

[Fri89] FRIEDMAN, J. H.: Regularized gaussian discriminant analysis. In: *Journal of the American Statistical Association* 84 (1989), S. 165–175

[HSV05] HAMMER, B.; STRICKERT, M.; VILLMANN, T.: Supervised neural gas with general similarity measure. In: *Neural Processing Letters* 21 (2005), February, 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*. 2nd. Berlin, Heidelberg : Springer, 1997

[KWR+01] KHAN, J.; WEI, J. S.; RINGNÏ¿$\frac{1}{2}$R, M. [u. a.]: Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. In: *Nature Medicine* 7 (2001), S. 673–679

[LCKL05] LEE, E. K.; COOK, D.; KLINKE, S.; LUMLEY, T.: Projection Pursuit for Exploratory Supervised Classification. In: *Journal of Computational and Graphical Statistics* 14 (2005), Nr. 4, S. 831–846

[NHBM98] NEWMAN, D. J.; HETTICH, S.; BLAKE, C. L.; MERZ, C. J. *UCI Repository of machine learning databases*. `http://archive.ics.uci.edu/ml/`, last visit 19.04.2008. 1998

[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.): *Proc. of European Symposium on Artifiacal Neural Networks (ESANN)*. Bruges, Belgium, April 2007, S. 37–42

[SBS+08a] SCHNEIDER, P.; BUNTE, K.; STIEKEMA, H.; HAMMER, B.; VILLMANN, T.; BIEHL, M.: *Regularization in Matrix Relevance Learning*. 2008. – submitted

[SBS+08b] SCHNEIDER, P.; BUNTE, K.; STIEKEMA, H.; HAMMER, B.; VILLMANN, T.; BIEHL, M.: *Regularization in Matrix Relevance Learning, Technical Report*. 2008. – submitted

[SRW+00] SCHERF, U.; ROSS, D. T.; WALTHAM, M. [u. a.]: A Gene Expression Database for the Molecular Pharmacology of Cancer. In: *Nature Genetics* 24 (2000), S. 236–244

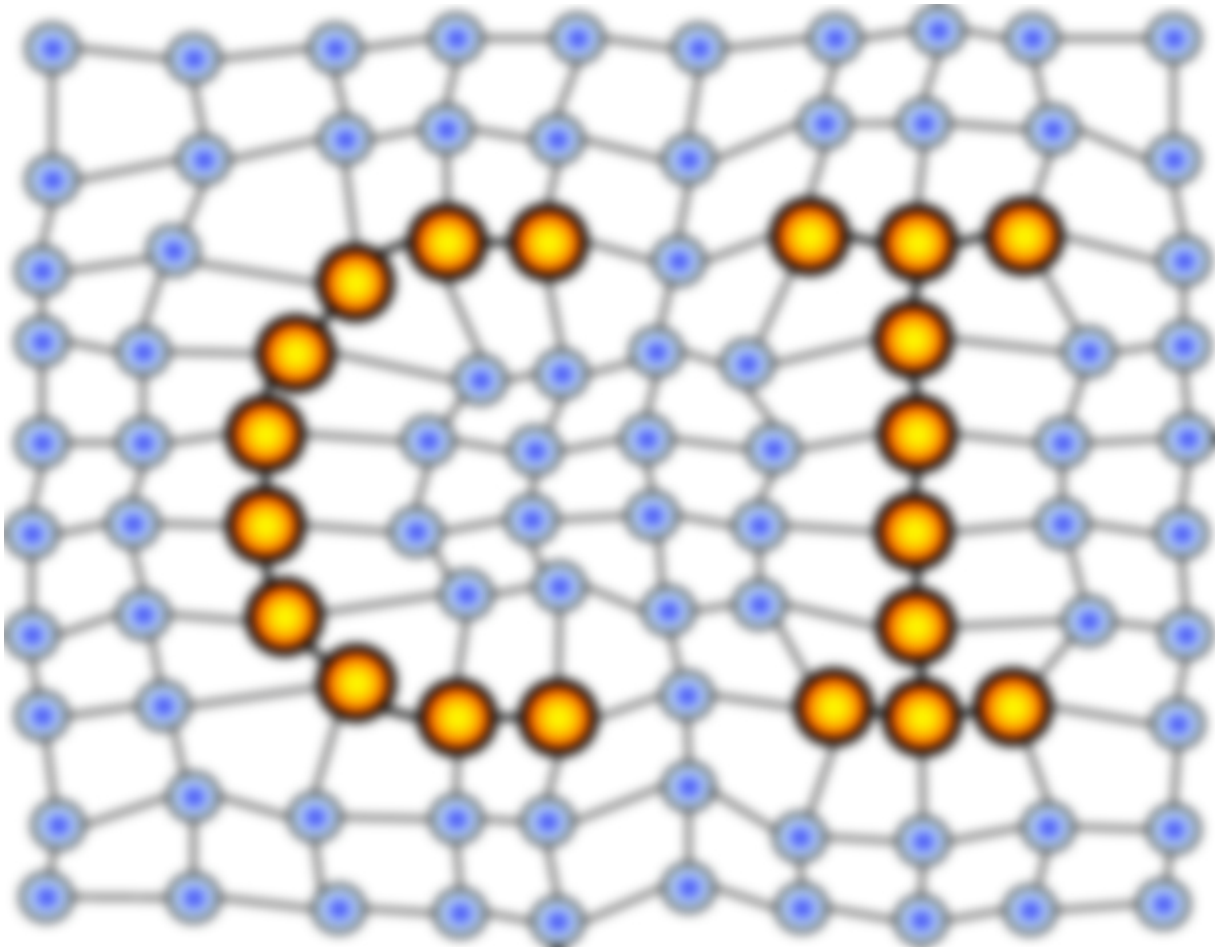
[SY96] SATO, A. S.; YAMADA, K.: Generalized learning vector quantization. In: *Advances in Neural Information Processing Systems* Bd. 8, 1996, S. 423–429

[WBS06] WEINBERGER, K. Q.; BLITZER, J.; SAUL, L. K.: Distance Metric Learning for Large Margin Nearest Neighbor Classification. In: *Advances in Neural Information Processing Systems* 18 (2006), S. 1473–1480

[ZZR04] ZHANG, L.; ZHANG, A.; RAMANATHAN, M.: VizStruct: exploratory visualization for gene expression profiling. In: *Bioinformatics* 20 (2004), S. 85–92

MACHINE LEARNING REPORTS

Report 03/2008



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.