Neural Networks 26 (2012) 159–173

Contents lists available at SciVerse ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Limited Rank Matrix Learning, discriminative dimension reduction and visualization

Kerstin Bunte [a,*], Petra Schneider [b], Barbara Hammer [c], Frank-Michael Schleif [c], Thomas Villmann [d], Michael Biehl [a]

[a] University of Groningen, Johann Bernoulli Institute for Mathematics and Computer Science, The Netherlands

[b] University of Birmingham, School of Clinical and Experimental Medicine, United Kingdom

[c] Bielefeld University, Center of Excellence-Cognitive Interaction Technology CITEC, Germany

[d] University of Applied Sciences Mittweida, Department of MPI, Germany

ARTICLE INFO

Article history:
Received 10 October 2010
Received in revised form 13 September 2011
Accepted 7 October 2011

Keywords:
Learning Vector Quantization
Classification
Visualization
Adaptive metrics
Dimension reduction

ABSTRACT

We present an extension of the recently introduced Generalized Matrix Learning Vector Quantization algorithm. In the original scheme, adaptive square matrices of relevance factors parameterize a discriminative distance measure. 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.

In particular, for very large dimensional data, the limitation of the rank can reduce computation time and memory requirements significantly. Furthermore, two- or three-dimensional representations constitute an efficient visualization method for labeled data sets. The identification of a suitable projection is not treated as a pre-processing step but as an integral part of the supervised training. Several real world data sets serve as an illustration and demonstrate the usefulness of the suggested method.

1. Introduction

Learning Vector Quantization (LVQ) (Kohonen, 2001) 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. (Biehl, Ghosh, & Hammer, 2007; Bojer, Hammer, Schunk, & von Toschanowitz, 2001; Bunte, Biehl, Petkov, & Jonkman, 2009; Bunte, Hammer, Schneider, & Biehl, 2009; Bunte, Hammer, Wismüller, & Biehl, 2010a; Hammer, Strickert, & Villmann, 2005a; Hammer & Villmann, 2002; Schneider, Biehl, & Hammer, 2009; Villmann, Merenyi, & Hammer, 2003). The method is easy to implement and its complexity is controlled by the user in a straightforward way. LVQ can be applied to multi-class problems without further complication and the resulting classifiers can be interpreted intuitively. This is due to the fact that 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 (1996) 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. Recently, also kernelized versions have been proposed (Schleif, Villmann, Hammer, Schneider, & Biehl, 2010). A key issue in all LVQ algorithms, with or without an 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 (Bojer et al., 2001; Hammer, Strickert, & Villmann, 2005b; Hammer & Villmann, 2002) which employ adaptive scaling factors for each dimension in feature space.

An important extension of this concept has been introduced in Schneider et al. (2009): in the so-called Generalized Matrix LVQ (GMLVQ) 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

* Corresponding author.
E-mail address: k.bunte@rug.nl (K. Bunte).
URL: http://www.cs.rug.nl/~kbunte/ (K. Bunte).

0893-6080/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2011.10.001

individual prototypes, are technically straightforward and allow for the parameterization of more complex decision boundaries.

Here we present and discuss an important modification: 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 twofold: (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.

Although unrestricted GMLVQ displays a tendency to reduce the rank of the relevance matrices in the training process, the advantages of restricting the rank explicitly are obvious. In particular for nominally very high-dimensional data, e.g. in image analysis or bioinformatics, unrestricted relevance matrices become intractable. In addition, optimization results can be poor when the search is performed in an unnecessarily large parameter space. Furthermore, the exact control of the rank allows for pre-defining the dimension of the intrinsic representation and is, for instance, suitable for the discriminative visualization of labeled data sets. 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. Visualization enables to use the astonishing cognitive capabilities of humans for visual perception when extracting information from large data volumes. Structural characteristics can be captured almost instantly by humans, independent of the number of displayed points. Classical unsupervised dimension reduction techniques represent data points contained in a high dimensional data manifold by low dimensional counterparts in, for instance, two or three dimensions, while preserving as much information as possible. Since it is not clear in advance which parts of the data are relevant to the user, this problem is inherently ill-posed: depending on the specific data domain and the situation at hand, different aspects can be in the focus of attention. Prior knowledge, in the form of label information, can be used to formulate a well-defined objective in terms of the classification performance.

There exist a few classical dimensionality reducing visualization tools which take class labels into account: Classical Fisher linear discriminant analysis (LDA), the recently introduced local Fisher discriminant analysis (LFDA) (Sugiyama & Roweis, 2007), Neighborhood Component Analysis (NCA) (Goldberger, Roweis, Hinton, & Salakhutdinov, 2004), as well as partial least squares regression (PLS) offer supervised linear visualization techniques. Kernel techniques extend these settings to nonlinear projections (Baudat & Anouar, 2000; Ma, Qu, & Wong, 2007). Adaptive dissimilarity measures which modify the metric used for projection according to the given auxiliary information have been introduced in Kaski, Sinkkonen, J, and Peltonen (2001), Peltonen, Klami, and Kaski (2004), and Bunte et al. (2010a). The resulting metric can be integrated into various techniques such as SOM, MDS, or a recent information theoretic model for data visualization (Kaski et al., 2001; Peltonen et al., 2004; Venna, Peltonen, Nybo, Aidos, & Kaski, 2010). An ad hoc metric adaptation is used in Geng, Zhan, and Zhou (2005) to extend Isomap (Tenenbaum, Silva, & Langford, 2000) to class labels. Alternative approaches change the cost function of dimensionality reduction, for instance by using conditional probabilities, class-wise similarity matrices or introducing a covariance-based coloring matrix for the side information as proposed in Iwata et al. (2007), Memisevic and Hinton (2005), and Song, Smola, Borgwardt, and Gretton (2008).

Before we describe our method more formally in Section 3 we review GMLVQ in the following section. In Section 4, we apply the novel LiRaM LVQ to a benchmark problem and study the influence of the dimension reduction on the classification performance. We also compare the limited rank version to the naive approach of taking the first components of the full rank GMLVQ. We show that reducing the rank after training not only requires more memory and CPU time, but also yields inferior classification performance compared to LiRaM LVQ. In Section 5 we present example applications of our algorithm in the visualization of labeled data. We also compare with visualizations obtained by LFDA and NCA. We conclude by summarizing our findings and providing an outlook on perspective investigations.

2. Review of Generalized Matrix LVQ

In this section we briefly review the Generalized Matrix LVQ algorithm (Schneider et al., 2009). We will assume that training is based on n examples of the form $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^N \times \{1, \ldots, C\}$, where N is the dimension of feature vectors and C is the number of classes. Learning Vector Quantization (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 $\boldsymbol{w}_i \in \mathbb{R}^N$ and the respective class label $c(\boldsymbol{w}_i) \in \{1, \ldots, C\}$. Given a distance measure $d^{\Lambda}(\boldsymbol{w}, \boldsymbol{x})$ in $\mathbb{R}^N$ parameterized by Λ, the classification is done according to a "winner takes all" or "nearest prototype" scheme: Any data point $\boldsymbol{x} \in \mathbb{R}^N$ is assigned to the class label $c(\boldsymbol{w}_i)$ of the closest prototype i with $d^{\Lambda}(\boldsymbol{w}_i, \boldsymbol{x}) \leq d^{\Lambda}(\boldsymbol{w}_j, \boldsymbol{x})$ 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 Sato and Yamada (1996) a very flexible approach is introduced, in which the training algorithm is guided by the minimization of a cost function

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

where the quantities

$$d_J^{\Lambda} = d^{\Lambda}(\boldsymbol{w}_J, \boldsymbol{x}_i) \quad \text{with } c(\boldsymbol{w}_J) = c(\boldsymbol{x}_i) \tag{2}$$

$$d_K^{\Lambda} = d^{\Lambda}(\boldsymbol{w}_K, \boldsymbol{x}_i) \quad \text{with } c(\boldsymbol{w}_K) \neq c(\boldsymbol{x}_i) \tag{3}$$

correspond to the distances of the feature vector $\boldsymbol{x}_i$ from the closest *correct (wrong)* prototype $\boldsymbol{w}_J(\boldsymbol{w}_K)$, 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 of a Mahalanobis distance

$$d^{\Lambda}(\boldsymbol{w}, \boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{w})^{\top} \Lambda (\boldsymbol{x} - \boldsymbol{w}) \tag{4}$$

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

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

and, hence

$$d^{\Lambda}(\boldsymbol{w}, \boldsymbol{x}) = [\Omega \ (\boldsymbol{x} - \boldsymbol{w})]^2 \tag{6}$$

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 an arbitrary matrix $\Omega \in \mathbb{R}^{N\times N}$. Gradients are evaluated with respect to the contribution of single instances $\boldsymbol{x}_i$ which are presented random sequentially. The algorithm has been introduced and discussed in Schneider et al. (2009) and will be modified in the following.

3. Limited Rank Matrix LVQ

In the following we extend the concept of GMLVQ to the use of rectangular matrices in the distance measure and refer to the corresponding algorithm as Limited Rank Matrix Learning Vector Quantization (LiRaM LVQ). We consider Ω to define a transformation from the original N-dimensional feature space to $\mathbb{R}^M$ with $M \leq N$ so that:

$$\Lambda = \Omega^\top \Omega \quad \text{with } \Omega \in \mathbb{R}^{M\times N}. \tag{7}$$

This section addresses the use of one global matrix for the dimension reduction and visualization. Modifications in the sense of extensions towards local distance measures will be discussed in the next section.

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 can identify a unique $\widehat{\Omega}$ by decomposing $\Lambda = \Omega^\top \Omega$ in a canonical way: We determine the normalized eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_M$ corresponding to the M ordered non-zero eigenvalues of Λ, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M$ and define $\widehat{\Omega}$ as:

$$\widehat{\Omega} = \left(\left[\sqrt{\lambda_1}\boldsymbol{v}_1, \sqrt{\lambda_2}\boldsymbol{v}_2, \ldots, \sqrt{\lambda_M}\boldsymbol{v}_M\right]\right)^\top. \tag{8}$$

In addition we choose the sign of v_i, such that the component of v_i with largest magnitude is positive. Note, that the value M limits the rank of the dissimilarity matrix Λ to a maximum of M. With the scheme Eq. (8) also a full matrix can be restricted after training. However, if eigenvectors with eigenvalues bigger than zero are omitted classification accuracy might get lost. We discuss this in Section 4.

Nominally, the matrix Ω will have more independent entries than the symmetric Λ whenever $M > (N+1)/2$. However, we have found no evidence that this ambiguity complicates the optimization problem. Therefore we consider throughout the following, general, unrestricted 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

$$\begin{aligned}\frac{\partial d_L^\Lambda}{\partial w_{L,r}} &= -2 \cdot \sum_n^N \sum_m^M \Omega_{mr}\Omega_{mn}(x_n - w_{L,n}) \\ &= -2\left[\Omega^\top \Omega\right]_r (\boldsymbol{x} - \boldsymbol{w}_L)\end{aligned} \tag{9}$$

$$\frac{\partial d_L^\Lambda}{\partial \boldsymbol{w}_L} = -2\Omega^\top \Omega(\boldsymbol{x} - \boldsymbol{w}_L) \tag{10}$$

$$\gamma^+ = \frac{\partial \mu}{\partial d_J^\Lambda} = \frac{2d_K^\Lambda}{(d_J^\Lambda + d_K^\Lambda)^2} \tag{11}$$

and

$$\gamma^- = \frac{\partial \mu}{\partial d_K^\Lambda} = \frac{-2d_J^\Lambda}{(d_J^\Lambda + d_K^\Lambda)^2}. \tag{12}$$

Here, $L \in \{J, K\}$ and the index J (K) refers to the closest correct (wrong) prototype $\boldsymbol{w}_J$ ($\boldsymbol{w}_K$) as introduced in Eq. (2).

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

$$\boldsymbol{w}_J^{\text{new}} = \boldsymbol{w}_J + \alpha_1 \cdot \gamma^+ \cdot 2\Lambda(\boldsymbol{x} - \boldsymbol{w}_J) \tag{13}$$

$$\boldsymbol{w}_K^{\text{new}} = \boldsymbol{w}_K + \alpha_1 \cdot \gamma^- \cdot 2\Lambda(\boldsymbol{x} - \boldsymbol{w}_K). \tag{14}$$

The corresponding matrix update reads

$$\begin{aligned}\frac{\partial d_L^\Lambda}{\partial \Omega_{mn}} &= 2\sum_i^N (x_n - w_{L,n})\Omega_{mi}(x_i - w_{L,i}) \\ &= 2\left[\Omega(\boldsymbol{x} - \boldsymbol{w}_L)\right]_m \cdot (x_n - w_{L,n})\end{aligned} \tag{15}$$

$$\frac{\partial \mu}{\partial \Omega_{mn}} = \left(\gamma^+ \frac{\partial d_J^\Lambda}{\partial \Omega_{mn}} + \gamma^- \frac{\partial d_K^\Lambda}{\partial \Omega_{mn}}\right)$$

$$\Omega_{mn}^{\text{new}} = \Omega_{mn} - \alpha_2 \cdot \frac{\partial \mu}{\partial \Omega_{mn}}. \tag{16}$$

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

$$\sum_i \Lambda_{ii} = \sum_{mn} \Omega_{mn}^2 = 1. \tag{17}$$

Note that the learning rates α_1 and α_2 can be chosen independently. In particular, we set $\alpha_1 \gg \alpha_2$ which implies that changes of the metric occur on a slower time scale than those of the prototypes. This setting has proven advantageous in many implementations of matrix relevance learning (Bojer et al., 2001; Hammer & Villmann, 2002; Schneider et al., 2009).

In all practical examples considered in the following, we apply a learning rate schedule of the form

$$\alpha_1(t) = \frac{\alpha_1^{\text{start}}}{1 + (t-1)\Delta\alpha_1} \tag{18}$$

and

$$\alpha_2(t) = \begin{cases} \dfrac{\alpha_2^{\text{start}}}{1 + (t - t_M)\Delta\alpha_2} & \text{for } t \geq t_M \\ 0 & \text{otherwise.} \end{cases} \tag{19}$$

Here, t corresponds to the current epoch, i.e. sweep through the data set, and $\alpha_{1,2}^{\text{start}}$ denotes the initial learning rates. Non-zero relevance updates are performed only after the first t_M epochs of prototype training. The computational costs scale linearly with the number of prototypes l, the dimension of the data N, the target dimension M and with the number of training examples n in each epoch $\mathcal{O}(lMNn)$. Initial positions $\boldsymbol{w}_i(t = 0)$ of the prototypes are determined by randomly selecting 1/3 of the available feature vectors in class $c(\boldsymbol{w}_i)$ 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. (17).

3.1. LiRaM LVQ with localized dissimilarities

For full rank matrices the Localized Generalized Matrix Learning Vector Quantization (LGMLVQ) was introduced and discussed in Schneider et al. (2009). It is based on the concept of localized matrices Ω_i individually adapted for each prototype or for each class, flexibly increasing the complexity of the LVQ system. The concept of LiRaM LVQ can also be expanded to the use of localized rectangular matrices, representing several local linear projections. The global combination of these local linear patches by means of charting is discussed in Brand (2003), and Bunte, Hammer, Wismüller, and Biehl (2010b).

In this contribution, we will investigate the use of localized matrices in combination with a global linear dimension reduction. This can be achieved by expanding the definition of the dissimilarity measure Eq. (4) with the combination of two matrices:

$$d_L^{2M}(\boldsymbol{w}_L, \boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{w}_L)^\top \Omega^\top \Psi_L^\top \Psi_L \Omega\ (\boldsymbol{x} - \boldsymbol{w}_L). \tag{20}$$

Here $\Omega \in \mathbb{R}^{M\times N}$ performs the dimension reduction with target dimension M, while the $\Psi_L \in \mathbb{R}^{M\times M}$ attached to the prototypes $\boldsymbol{w}_L$ define a localized dissimilarity measure in the transformed space. Consequently the visualizations correspond to nonlinear rather than piecewise linear decision boundaries in the M-dimensional space. In the experiments we also used class-wise dissimilarities Ψ_c with $c \in \{1, \ldots, C\}$ attached to the prototypes $\boldsymbol{w}_L$ with label $c(\boldsymbol{w}_L) = c$, which may be interesting in a setting with more than one prototype per class. In the following we will address this algorithm as Localized LiRaM LVQ (LLiRaM LVQ).

The prototype update reads:

$$\frac{\partial d_L^{2M}}{\partial \boldsymbol{w}_L} = -2\Omega^\top \Psi_L^\top \Psi_L \Omega (\boldsymbol{x} - \boldsymbol{w}_L)^\top \tag{21}$$

$$\frac{\partial \mu}{\partial \boldsymbol{w}_L} = \gamma_2^L \cdot \frac{\partial d_L^{2M}}{\partial \boldsymbol{w}_L} \tag{22}$$

$$\boldsymbol{w}_L^{\text{new}} = \boldsymbol{w}_L + \alpha_1 \cdot \frac{\partial \mu}{\partial \boldsymbol{w}_L} \tag{23}$$

with $L \in \{J, K\}$ and

$$\gamma_2^J = \frac{\partial \mu}{\partial d_J^{2M}} = \frac{2 d_K^{2M}}{(d_J^{2M} + d_K^{2M})^2}, \tag{24}$$

$$\gamma_2^K = \frac{\partial \mu}{\partial d_K^{2M}} = \frac{-2 d_J^{2M}}{(d_J^{2M} + d_K^{2M})^2}. \tag{25}$$

Furthermore, we obtain

$$\frac{\partial d_L^{2M}}{\partial \Omega} = 2 \cdot \Psi_L^\top \Psi_L \Omega (\boldsymbol{x} - \boldsymbol{w}_L)(\boldsymbol{x} - \boldsymbol{w}_L)^\top \tag{26}$$

$$\frac{\partial \mu}{\partial \Omega} = \gamma_2^K \cdot \frac{\partial d_K^{2M}}{\partial \Omega} + \gamma_2^J \cdot \frac{\partial d_J^{2M}}{\partial \Omega} \tag{27}$$

$$\Omega^{\text{new}} = \Omega - \alpha_2 \cdot \frac{\partial \mu}{\partial \Omega}. \tag{28}$$

The localized dissimilarities Ψ_L are updated according to:

$$\frac{\partial d_L^{2M}}{\partial \Psi_L} = 2 \cdot \Psi_L \Omega (\boldsymbol{x} - \boldsymbol{w}_L)(\boldsymbol{x} - \boldsymbol{w}_L)^\top \Omega^\top \tag{29}$$

$$\frac{\partial \mu}{\partial \Psi_L} = \gamma_2^L \cdot \frac{\partial d_L^{2M}}{\partial \Psi_L} \tag{30}$$

$$\Psi_L^{\text{new}} = \Psi_L - \alpha_3 \cdot \frac{\partial \mu}{\partial \Psi_L}. \tag{31}$$

The matrix Ω can be used to transform the data points and prototypes into a lower dimensional space. In the transformed space the prototypes and the local matrices Ψ_L define the nonlinear decision boundaries. In Section 5 we will show some example visualizations of the LLiRaM LVQ.

4. A classification problem

As an illustrative example, we study the performance of the LiRaM LVQ algorithm on the image segmentation data set as provided in the UCI repository (Asuncion, Newman, Hettich, Blake, & Merz, 1998).

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 Asuncion et al. (1998) 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 LiRaM LVQ 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 (18), 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 (Schneider et al., 2009).

4.1. Performance dependence on M

We first study the simplest GMLVQ classifiers with only one prototype per class. For several values of M, we perform LiRaM LVQ on the given training set of 210 example data and observe the evolution of training and test accuracies with the number of epochs. In order to obtain reliable results and as an indication of the robustness and convergence properties, we present averages and standard deviations with respect to 10 different random initializations of the prototypes and matrix Ω.

Fig. 1 shows averaged learning curves for the example cases $M = 2$ and $M = 16$. We display the training and test accuracies averaged over 10 random initializations of the algorithm and the estimates of the corresponding standard errors are on the order 0.01 for $M = 2$ and below 0.005 for $M = 16$. Note that training and test accuracies can display a weak maximum in the course of learning. Therefore, for each M, we determine the number of epochs that yield the best mean training accuracy and display the corresponding test accuracy in the right panel of Fig. 1. The non-monotonic behavior could be cured by means of a proper regularization of GMLVQ by adding a penalty term to the cost function, see Schneider, Bunte, Hammer, and Biehl (2010). The additional application of this technique effects that the eigenvalues of Λ converge to $\frac{1}{M}$. Hence, the regularization prevents the algorithm from oversimplifying the classifier, and the computation of distance values is finally based on M features. Here, we resort to the above described early stopping technique for simplicity. We would like to point out that it relies only on the observed training accuracy and does not make use of test set information.

Fig. 1 also displays the relevance matrices and their eigenvalue spectra corresponding to the early stopping performances. In the case $M = 16$ we observe that only about 9–10 eigenvalues remain significantly different from zero. Even GMLVQ with unrestricted rank results in an effective low-dimensional representation of the data. One would expect that LiRaM LVQ with large enough M already yields the same performance as the unrestricted variant. Fig. 2 shows that this is indeed the case. 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. Many nonlinear

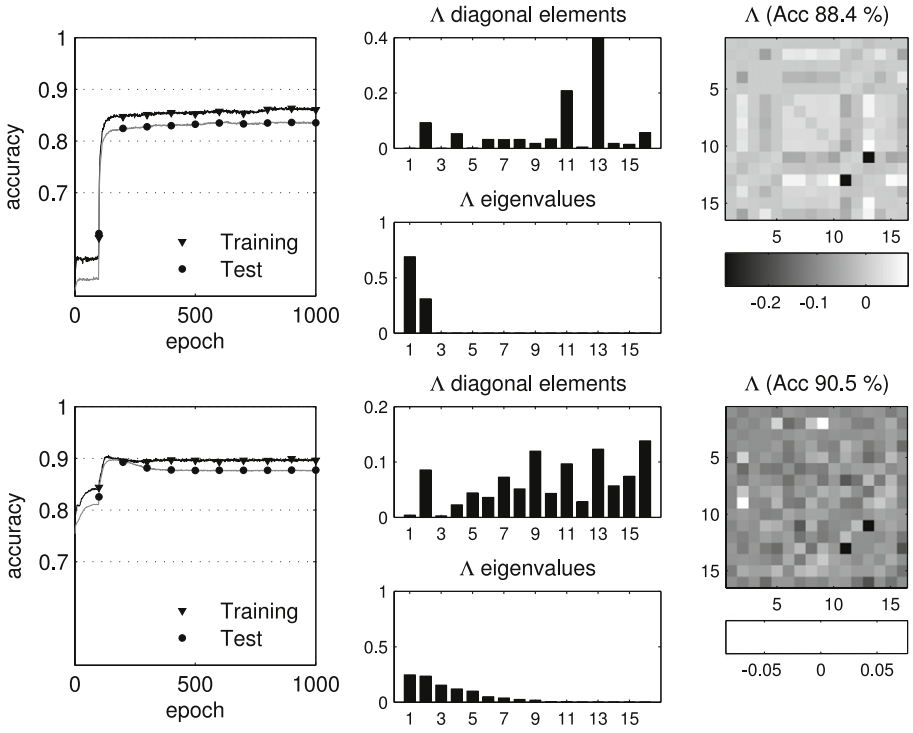


Fig. 1. Left panels: learning curves of LiRaM LVQ with one prototype per class for $M = 2$ (top) and $M = 16$ (bottom) when applied to the UCI image segmentation data set. Right panels: diagonal elements, eigenvalues and off-diagonal elements of the matrix Λ as obtained in a single run. The diagonal elements are set to zero for the plots.

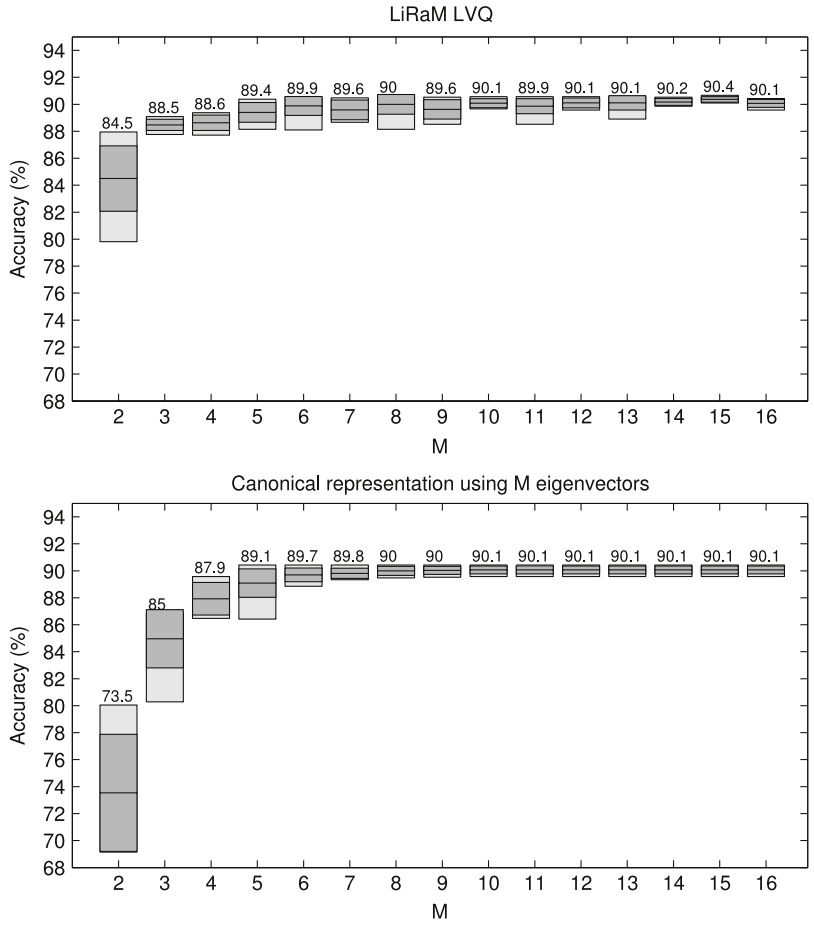


Fig. 2. Performance of the LiRaM LVQ (upper panel) and GMLVQ with successive matrix reduction following Eq. (8) (lower panel) using one prototype per class as a function of M for the UCI image segmentation data set. We display the test accuracy on average over 10 random initializations, also given as a numerical value. The light shading corresponds to the interval from worst to best accuracy, the darker area marks the standard deviations.

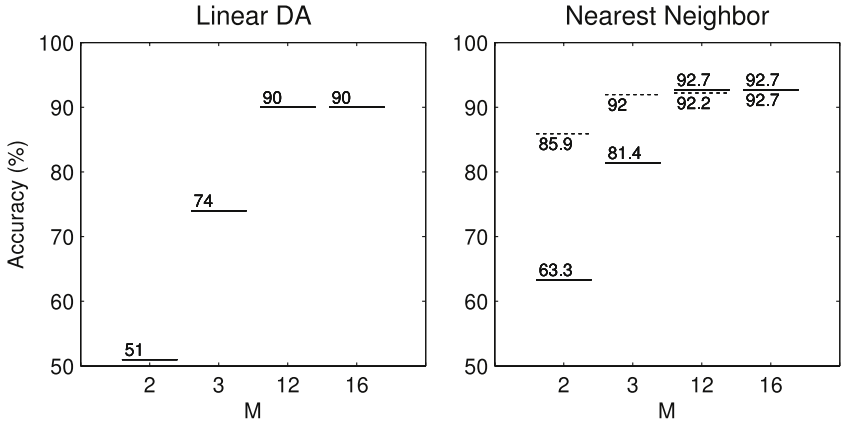


Fig. 3. UCI image segmentation data set. Left panel: test accuracy obtained by LDA as described in the text. Right panel: test accuracies for the Nearest-Neighbor classifier using the PCA-based transformation to M dimensions (solid lines). In addition, the results after transforming the data with Ω as obtained in LiRaM LVQ, the dotted lines mark the average over 10 random initializations as in Fig. 2.

dimension reduction methods such as t-distributed Stochastic Neighbor Embedding (t-SNE) do not lead to a unique solution, a data set may be visualized differently by the same technique in different runs. It can be argued (see e.g. van der Maaten & Hinton, 2008) that this effect is desirable since it mirrors different possible views of the given data and the ill-posedness of the problem of dimension reduction. Auxiliary information in the form of class labels can be useful to shape the problem in such settings and to resolve (parts of) the ambiguities inherent in the problem. However, if the intrinsic dimension of the data is larger than the target dimension some ambiguities may not be resolved.

Additionally, we investigate the performance of the full matrix system reducing the rank after training with Eq. (8) using only the first M eigenvalues and eigenvectors. The lower panel of Fig. 2 shows the test accuracies using the $M = 16$ matrices and the canonical representation with M eigenvectors for different values of M. As observed before, keeping less than the 5 eigenvalues in the successive restricted GMLVQ (lower panel of Fig. 2) results in a decrease of the classification accuracy. The drop in accuracy is especially significant when eigenvectors with relatively large eigenvalues are omitted. Just using the eigenvectors of the two largest eigenvalues for example shows a mean test accuracy which is 11% smaller than the corresponding LiRaM LVQ result for $M = 2$. Despite the computation time and memory efficiency, the limited rank version yields better preservation of the classification performance in the restricted setting than the heuristic dimension reduction after training omitting eigenvectors with eigenvalues significantly different from zero.

4.2. Comparison with other methods

Here we compare the LiRaM LVQ scheme with frequently used standard procedures of comparable complexity. Note that the complexity of LiRaM LVQ can be easily adapted by the number of prototypes. GMLVQ with only one prototype per class appears to be similar in spirit to the well known Linear Discriminant Analysis (LDA) (Bensmail & Celeux, 1996; Duda, Hart, & Stork, 2000; Friedman, 1989). In this method, a Multivariate Normal density (MVN) is fitted to the observed data in each class and 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 (Duda et al., 2000). The well known Nearest-Neighbor (1-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 (Duda et al., 2000). 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 LiRaM LVQ, 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 1-NN classifier. In the case $M = 16$, the full data set is employed without performing a PCA.

In Fig. 3, 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 1-NN is comparable to that of the LiRaM LVQ prescription. This finding is consistent with the M-dependence discussed in the previous section.

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 LiRaM LVQ scheme even with only 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 LiRaM LVQ, (see Eq. (6)). Now the simple 1-NN system performs significantly better, as displayed in the left panel of Fig. 3. The idea of determining a discriminative transformation directly within the KNN classification scheme has been put forward in Weinberger, Blitzer, and Saul (2006), there without considering dimension reduction. A more detailed comparison of Large Margin Nearest Neighbor (LMNN) with LiRaM LVQ is given in Bunte, Biehl, Jonkman, and Petkov (2011).

LiRaM LVQ with several prototypes per class and a global relevance matrix can implement piecewise linear decision boundaries, the complexity of which can exceed that of LDA or similar methods significantly. In previous applications of unrestricted GMLVQ to the UCI image segmentation data it has proven advantageous to assign 3 prototypes to class 5 (window) and 2 prototypes to all other classes. Fig. 4 shows that this setting improves the classification accuracies in comparison to the above studied case of a single prototype per class, cf. Fig. 2. As expected, the improvement is particularly pronounced for small M. In Fig. 5 we visualize typical properties of the relevance matrices obtained in the extreme cases $M = 2$ and $M = 16$. Note that even the unrestricted matrix

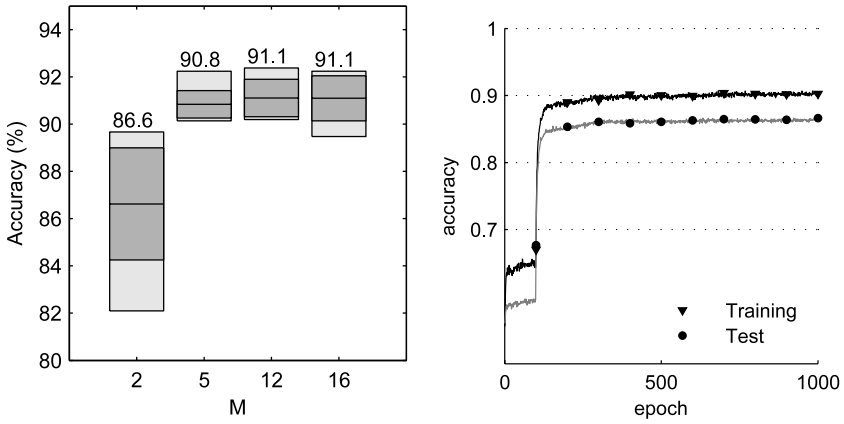
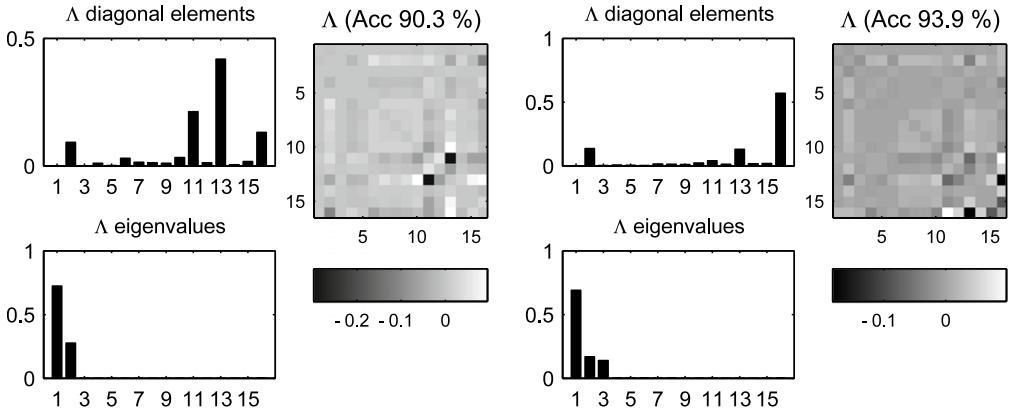


Fig. 4. UCI segmentation. Left panel: test accuracies achieved by LiRaM LVQ with 2 prototypes per class (3 in class 5) for different values of M; other details as in Fig. 2. Right panel: the corresponding learning curves for $M = 2$, i.e. mean training and test accuracy vs. the number training epochs.



(a) $M = 2$. (b) $M = 16$.

Fig. 5. Diagonal elements, eigenvalues, and off-diagonal elements of an example relevance matrix in LiRaM LVQ with two prototypes per class and three in class 5. Other details as in Fig. 1, right panels. The diagonal elements are set to zero for the plots.

displays only three non-zero eigenvalues. The increased complexity due to the larger number of prototypes facilitates good performance in spite of a very simple implicit representation of the data. The use of more eigendirections could be enforced by means of a matrix regularization scheme suggested in Schneider et al. (2010). We will address this issue in forthcoming studies.

5. Visualization of classification schemes

The LiRaM LVQ 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. Supervised dimension reduction has drawn some attention recently and some of the methods have been mentioned in the introduction. We explain two of these methods in the next section in more detail and will compare example visualizations of different data sets thereafter.

5.1. Local Fisher Discriminant Analysis

A supervised linear dimension reduction technique named Local Fisher Discriminant Analysis (LFDA) (Sugiyama & Roweis, 2007) was recently introduced as a combination of the well known Fisher Discriminant Analysis (FDA) (Fisher, 1936) and the unsupervised Locality-Preserving Projection (LPP) (He & Niyogi, 2003). FDA works particularly well, when each class can be modeled as a unimodal Gaussian. It is based on the within-class and between-class scatter matrix and finds a transformation matrix T, such that the between-class scatter is maximized, while the within-class scatter is minimized. This optimization problem can be solved by means of a generalized eigenvalue problem (Fukunaga, 1990). Furthermore the between-class scatter matrix has a rank limited to the number of classes minus one ($c - 1$). This implies that FDA can find at most $c - 1$ meaningful features, which constitutes a serious restriction in practice. LPP on the other hand is an unsupervised dimension reduction technique based on pairwise affinities $A_{i,j} \in [0, 1]$ between data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The aim is to find a transformation matrix T such that local neighborhoods are preserved in the embedding space.

The LFDA efficiently combines the ideas of both methods and facilitates the dimension reduction of multi-modal labeled data by maximizing the between-class separability, while preserving the local structure within classes. The local within-class and local between-class scatter matrices $S^{(w)}$ and $S^{(b)}$ are defined using pairwise affinities of the data:

$$S^{(w)} = \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}^{(w)}(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^{\top} \tag{32}$$

$$S^{(b)} = \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}^{(b)}(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^{\top}, \tag{33}$$

where n denotes the number of samples and

$$W_{i,j}^{(w)} = \begin{cases} A_{i,j}/n_l & \text{if } y_i = y_j = l \\ 0 & \text{if } y_i \neq y_j \end{cases} \tag{34}$$

$$W_{i,j}^{(b)} = \begin{cases} A_{i,j}\left(\frac{1}{n} - \frac{1}{n_l}\right) & \text{if } y_i = y_j = l \\ 1/n & \text{if } y_i \neq y_j. \end{cases} \quad (35)$$

The value n_l denotes the number of samples from class l. Therefore, LFDA aims at finding a transformation matrix T, such that nearby data pairs of the same class are also close in the embedding and data points of different classes are separated from each other. Similar to FDA and LFDA, a projection can be computed analytically by solving a generalized eigenvalue problem:

$$T = \underset{T \in \mathbb{R}^{N \times M}}{\operatorname{argmax}} \left[tr\left((T^\top S^{(w)} T)^{-1} T^\top S^{(b)} T \right) \right]. \quad (36)$$

In contrast to FDA the LFDA does not have the same rank limitation. Therefore a dimension reduction to arbitrary dimensions is possible. However, the embedding crucially depends on the computation of the pairwise affinities. In Sugiyama and Roweis (2007) four definitions of the affinity matrix are given. In the following experiments we use the "local scaling" method, which is also used in the provided implementation.[1] Here the density of the data is taken into account in a heuristic manner: a local scaling based on the k-th nearest neighbor is included. In the experiments we tried different values of k to find good visualizations.

5.2. Neighborhood Component Analysis

Recently, a supervised dimension reduction method called NCA has been introduced (Goldberger et al., 2004). It aims in the maximization of the expected number of correctly classified samples by a stochastic variant of the nearest neighbor classifier. Therefore, NCA seeks a transformation matrix T_{NCA} such that the between-class separability is maximized:

$$T_{\mathrm{NCA}} = \underset{T \in \mathbb{R}^{N \times M}}{\operatorname{argmax}} \left(\sum_{i=1}^{n} \sum_{y_j = y_i} p_{i,j} \left(TT^\top \right) \right), \quad (37)$$

where

$$p_{i,j}(U) = \begin{cases} \dfrac{\exp\{-(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top U (\boldsymbol{x}_i - \boldsymbol{x}_j)\}}{\sum_{k \neq i} \exp\{-(\boldsymbol{x}_i - \boldsymbol{x}_k)^\top U (\boldsymbol{x}_i - \boldsymbol{x}_k)\}} & \text{if } i \neq j \\ 0 & \text{if } i = j. \end{cases}$$

Thus, similar to LFDA, nearby data pairs from the same class should be close in the embedding space. This ensures that also multi-modal structure of the data can be preserved. However, the optimization problem is non-convex and there is no guarantee that the global optimum can be obtained. The optimization was proposed as a gradient ascent method and we use the provided implementation[2] for the experiments. Note, that NCA needs to compute the pairwise dissimilarities between samples of the same class in every step. Although LiRaM LVQ also follows a gradient procedure it computes only the dissimilarities with respect to the prototypes in every step. Since the number of prototypes per class is usually much smaller than the number of samples, the computational costs per gradient step are significantly lower than for NCA. In the implementation a Polack–Ribiere flavor of conjugate gradients is used to compute search directions, and a line search using quadratic and cubic polynomial approximations. There is mainly one parameter to change: l the length of the run. It corresponds to the maximum number of line searches.

[1] MATLAB implementation LFDA: http://sugiyama-www.cs.titech.ac.jp/~sugi/software/LFDA/

[2] MATLAB implementation for NCA: http://www.ics.uci.edu/~fowlkes/software/nca/

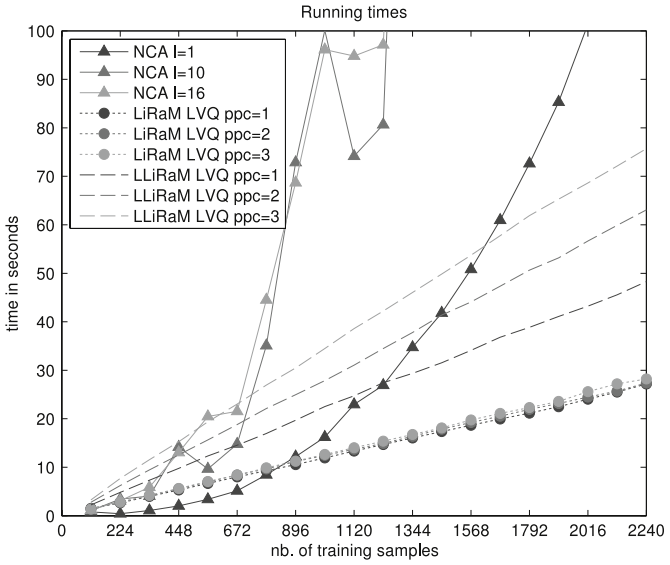


Fig. 6. Runningtimes of NCA and the LiRaM LVQ variants in dependence of the number of training samples. Details can be found in the text.

Fig. 6 displays the running times of NCA and the proposed LVQ variants in dependence on the number of training samples. We run the experiments on the same machine[3] using Matlab implementations. We used differently sized subsets of the seven class segmentation data set, which is investigated in the following section and reduce the dimension to $M = 2$. Furthermore we compare different parametrizations of the models. For NCA we show the running times for different numbers of line searches l and for the LVQ variants we vary the number of prototypes per class abbreviated by "ppc". It can be seen that the computation time grows linearly with the number of training samples using the LVQ approaches, while the complexity of NCA grows quadratically.

5.3. The segmentation data set

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. 2 (single prototype per class) and Fig. 4 (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 GMLVQ system, see Figs. 2 and 4 for the range of observed accuracies. With a single prototype per class, a maximum classification accuracy of 88.4% 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. The use of several prototypes with LLiRaM LVQ enhances the accuracy by realizing more complex piecewise linear decision boundaries.

Furthermore we trained the LLiRaM LVQ under the same conditions ten times on the training set of the segmentation data and used the resulting transformations and prototypes to visualize the data. The run showing the best performances is shown in Fig. 7 with the quality given in Table 1. The mean accuracy over all runs on the training data is 85% with a standard deviation (STD) of 0.04 with one prototype per class and class-wise dissimilarities Ψ_c. LLiRaM LVQ implements nonlinear decision boundaries, which show already good accuracies using one prototype per class. With this particular data set, using more prototypes does not improve the classification significantly.

[3] Quadcore: Intel(R) Core(TM) i5 CPU 750 @ 2.67 GHz.

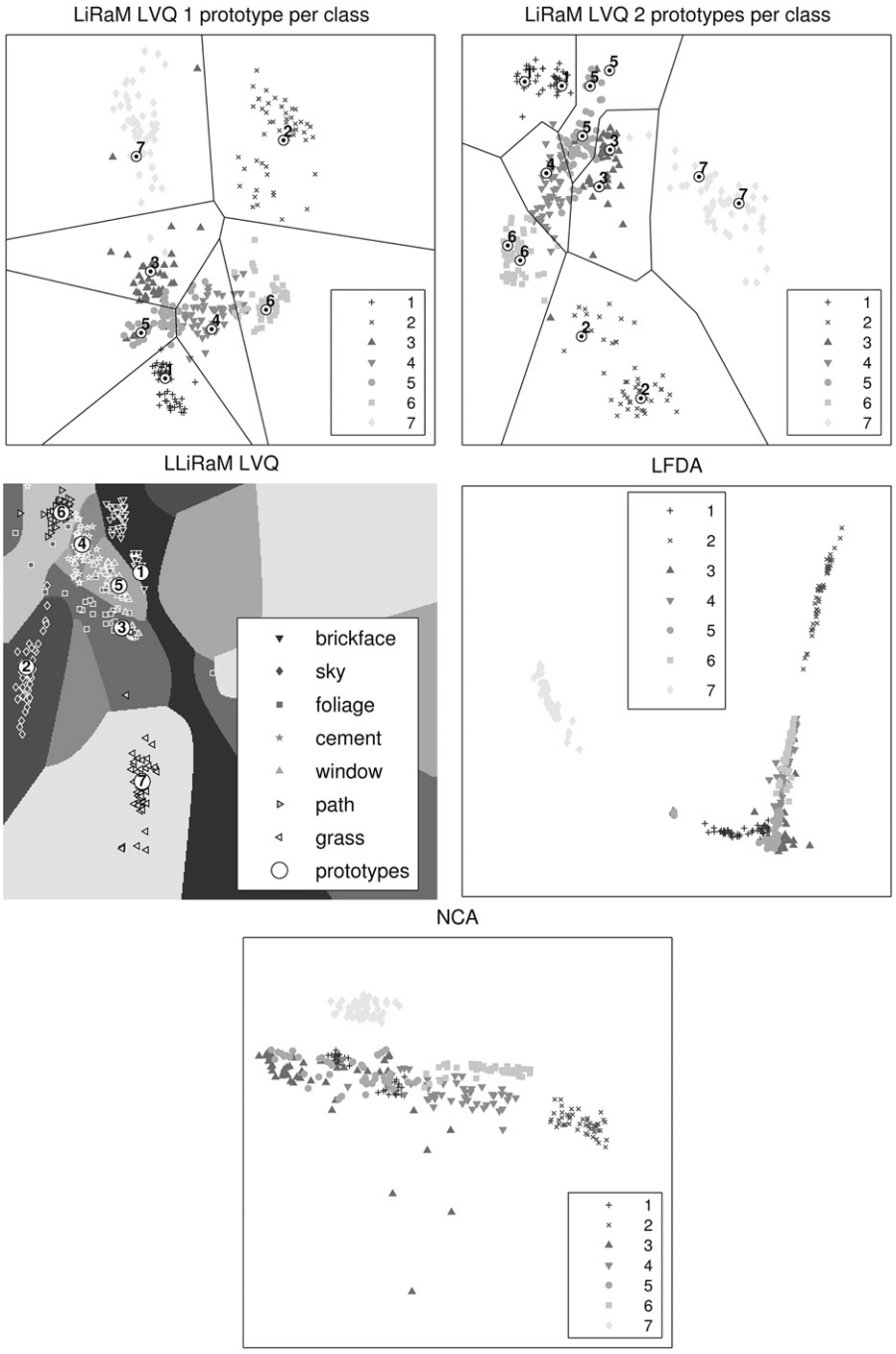


Fig. 7. Visualizations of the UCI segmentation data set acquired by the different methods. For the sake of clarity we display only 50 examples per class. A detailed explanation can be found in the text.

Next, we employ the implementation of LFDA and NCA from the original authors with default parameters and tried a range of k and $l \in [1, 30]$. We observed that both methods crucially depend on the parameter used. The accuracy of the training set measured by an 1-NN classification on the embedding acquired by LFDA, for example, ranges from the best accuracy of 83.7% with $k = 2$ and the worst accuracy of 66.6% with $k = 25$. For NCA the worst accuracy of 56.2% is observed with $l = 1$ and with $l \geq 16$ the training accuracy reaches 90%. The number of protoytpes and the initialization in the LiRaM LVQ setting is less crucial with respect to the classification accuracy.

Fig. 7 displays the visualizations with best classification performance on the segmentation data set acquired by the different techniques explained above. This multi-class problem allows for very good classification performance already in two dimensions. The localized variant of LiRaM LVQ can realize more complicated non-linear decision boundaries than the global version. However, overfitting effects become possible: For one prototype per class we observe an improvement although empty cells appear in the tessellation. With two prototypes per class no further improvement is observed. In all visualizations the classes "sky" and "grass" can be separated quite well. For the other classes the visualizations differ in arrangement and shape of the clusters. The LiRaM LVQ visualizations show equal or superior quality compared to the other methods. An overview of the visualization quality of the different methods on the data sets can be found in Table 1. The classification accuracy in the original space is usually larger, than the accuracy in the low-dimension space after

Table 1
Classification and 1-NN accuracies (in %) on the visualizations of the data sets.

Method/data set	acc. training	acc. test
Segmentation data		
LiRaM LVQ 7P (classification accuracy)	92.9	88.0
LiRaM LVQ 7P (1-NN acc. on embedding)	85.7	87.0
LiRaM LVQ 14P (classification accuracy)	91.9	90.3
LiRaM LVQ 14P (1-NN acc. on embedding)	88.6	87.5
LLiRaM LVQ (classification accuracy)	89.0	85.7
LLiRaM LVQ (1-NN acc. on embedding)	88.6	87.4
LFDA (1-NN acc. on embedding)	83.7	85.8
NCA (1-NN acc. on embedding)	90.0	87.1
Colorado data 2D		
LiRaM LVQ (classification accuracy)	83.0	80.0
LiRaM LVQ (1-NN acc. on embedding)	79.6	84.6
LLiRaM LVQ (classification accuracy)	78.7	73.8
LLiRaM LVQ (1-NN acc. on embedding)	79.9	83.7
LFDA (1-NN acc. on embedding)	50.4	61.1
NCA (1-NN acc. on embedding)	81.5	89.7
Colorado data 3D		
LiRaM LVQ (classification accuracy)	88.9	86.3
LiRaM LVQ (1-NN acc. on embedding)	93.3	96.4
LLiRaM LVQ (classification accuracy)	87.7	85.8
LLiRaM LVQ (1-NN acc. on embedding)	92.8	96.1
LFDA (1-NN acc. on embedding)	89.6	93.8
NCA (1-NN acc. on embedding)	92.6	95.5

transformation. However, the numbers show that in most cases the supervised dimension reduction was able to preserve high accuracies even in the reduced spaces. We would like to point out once more, that the computational effort for NCA is much larger than for the LiRaM LVQ variants. NCA computes all pairwise distances, while the LVQ approaches are based on a small number of prototypes. In particular, for large data sets the computational effort may be reduced significantly compared to NCA.

5.4. High dimensional gene expression data

Discriminative visualization can be particularly useful in the context of medical data. Here we apply the LiRaM LVQ algorithm to two gene expression data sets which were recently analyzed by Faith, Mintram, and Angelova (2006).

The first set concerns *small round blue cell childhood tumors*, and we refer to it as SRBCT (Faith et al., 2006). It comprises cDNA microarray expression levels of 50 pre-selected genes in 83 different samples (Khan et al., 2001). 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 (Scherf, Ross, & Waltham, 2000). 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 Faith et al. (2006) 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) (Ewing & Cherry, 2001), VizStruct (Zhang, Zhang, & Ramanathan, 2004), a dendrogram based method (Eisen, Spellman, Brown, & Botstein, 1998), and Projection Pursuit (Lee, Cook, Klinke, & Lumley, 2005). 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 publicly available (Faith et al., 2006).

First, we apply LiRaM LVQ with one prototype per class to the SRBCT data set. Results presented here 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 of the system. A typical outcome is displayed in Fig. 8 (top row left panel) where the obtained 2*D* visualization perfectly separates the four classes. Error free visualizations were also obtained by Faith et al., see Faith et al. (2006) for comparison.

The analogous application of LiRaM LVQ to the NCI 8-class-problem shows a 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. 8 (upper row, right panel) displays a perfectly separating visualization.

For the sake of completeness we show the error-free example results of the LLiRaM LVQ with one prototype per class in Fig. 8 (bottom row). The algorithm was trained with the same parameters as the global version on both, the whole SRBCT and NCI, data set. Again the four-class problem SRBCT can be separated in every run with random initialization, whereas the training on the NCI data set shows some variation in classification accuracy. We achieved on the NCI data a mean average accuracy of 94.6% with a standard deviation of 0.02 over the 10 random initializations.

The visualizations of these data sets achieved by LFDA and NCA are shown in Fig. 9. LFDA was performed on the SRBCT data set with $k \in [1, 10]$, all yielding error free visualizations. On the NCI data set the accuracy varied from 91.8% achieved with $k = 4$ to the best accuracy of 96.7% using $k = 1$. For the training of NCA on the SRBCT data set with l varying from one to 10, we observed error free visualizations for $l \geq 3$ and the worst accuracy of 80.7% for $l = 1$. On the NCI data set an error free visualization is found for $l \geq 10$ and the worst performance was 59% observed with $l = 1$.

In Faith et al. (2006), 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 Faith et al. (2006), which display either slightly overlapping classes or only very small gaps between some of them. Other methods considered in Faith et al. (2006) 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 at comparably low computational costs.

5.5. Satellite remote sensing data

Here we apply the algorithm to a large real world data set: a multi-spectral satellite image of the Colorado area, focusing on visualizing the class structure. Remote sensing spectral images consist of an array of multi-dimensional vectors (spectra) assigned to particular spatial regions (pixels) reflecting the response of a spectral sensor at various wavelengths. A spectrum is a characteristic pattern that provides a clue to the surface material within the respective area. The use of these data includes areas such as mineral exploration, land use, forestry; and many other activities of economic significance.

We consider a data set that corresponds to an image taken close to Colorado Springs using satellites of the LANDSAT-TM type. The size of the image is 1907 × 1784 pixels, each of which correspond to an area of 900 m^2 on the ground. The spectrum is represented by a 6-dimensional feature vector. The aim of the classification is to assign each pixel to one of 14 classes, corresponding to specific surface covers such as different types of forests, alpine vegetation, water, etc., (see Hammer and Villmann (2002) and Villmann et al. (2003) for a detailed description and Table 2 for the list of classes).

A labeling of the entire image was provided by experts and serves as the target classification. For further details of the data set we refer the reader to Hammer and Villmann (2002), and Villmann et al. (2003) where the authors apply scaled Euclidean distance in combination with a Growing Self-Organized Map (GSOM). Test

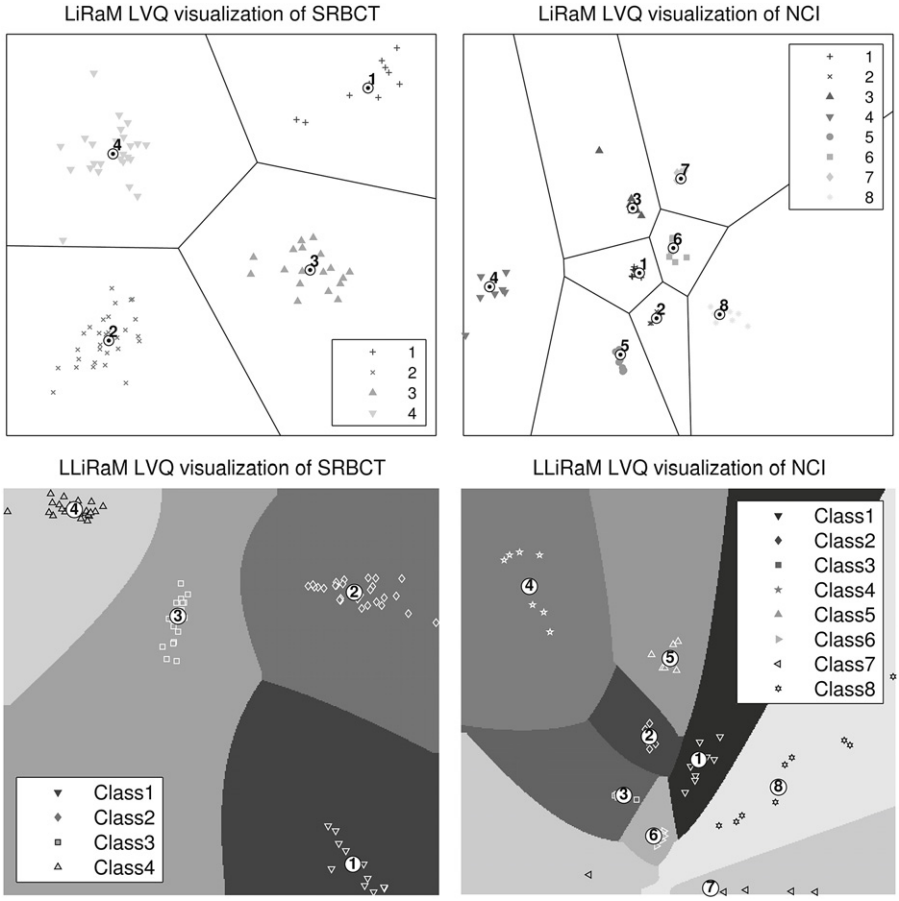


Fig. 8. Two-dimensional visualizations of the SRBCT data set (left column) and the NCI data (right column) obtained by the different variants of LiRaM LVQ explained in the text.

Table 2
Short description of the different classes of the satellite image and the number of pixels in each class.

Class	Ground cover type	# pixels
1	Scotch pine	581 424
2	Douglas fir	355 145
3	Pine/fir	181 036
4	Mixed pine forest	272 282
5	Supple/prickle pine	144 334
6	Aspen/mixed pine forest	208 152
7	Without vegetation	170 196
8	Aspen	277 778
9	Water	16 667
10	Moist meadow	97 502
11	Bush land	127 464
12	Grass/pastureland	267 495
13	Dry meadow	675 048
14	Alpine vegetation	27 556
0	Not classified	9

accuracies in the range of 90% have been achieved depending on the specific method in use.

For the following, we selected 2000 examples per class randomly, used as a training set. We also give the accuracies evaluated with respect to the whole data set of 3 402 088 data points. We have performed 10 runs of LiRaM LVQ with $M = 2, 3$ and three prototypes per class. After 1500 training epochs we observe only very little variation due to the random initialization of the system. The range of training accuracies is 79.8%–83% for $M = 2$ and 87.5%–88.9% for $M = 3$, respectively. The classifiers with the best training set performance achieve accuracies on the whole set of 80.1% ($M = 2$) and 86.3% ($M = 3$), see Table 1. In spite of the low-dimensional representation and the relatively small numbers of prototypes we achieve very good accuracies. This is consistent with the analysis in Villmann et al. (2003) which suggests that good classification performance requires at least a two- or three-dimensional representations of the data.

Here, we are mainly interested in the discriminative visualization of the data set. Fig. 10 shows the data globally projected into two and three dimensions, respectively. We also trained the localized LiRaM LVQ on 2000 random samples from each class with slightly different parameters: 300 epochs, learning rates beginning with $\alpha^{\text{start}} = 0.001$ and $\Delta\alpha = 0.0001$ for the prototypes, the matrix Ω and the class-wise matrices Ψ_c respectively. We trained the system with two and three prototypes per class. The average accuracy on the training data is 75% with STD 0.03 in the two-dimensional case with 28 prototypes. In three dimensions with three prototypes per class we obtain a mean accuracy of 85.2% and STD 0.02. These results correspond to the findings in Hammer and Villmann (2002) where Generalized Relevance Learning Vector Quantization (GRLVQ) was applied to the data set: When pruning to three dimensions a classification performance of ca. 84% can be achieved, while dropping further dimensions decreases the accuracy significantly. The visualizations resulting from the best run in two and three dimensions are shown in Fig. 10 (bottom row). Furthermore, the confusion matrix for the three-dimensional case containing information about the class-wise accuracies and misclassification can be found in Table 3. We also provide the original labeling of the satellite image and the estimated Labels with misclassification. The corresponding graphics can be found in Fig. 11. The projections facilitate a detailed interpretation and analysis of

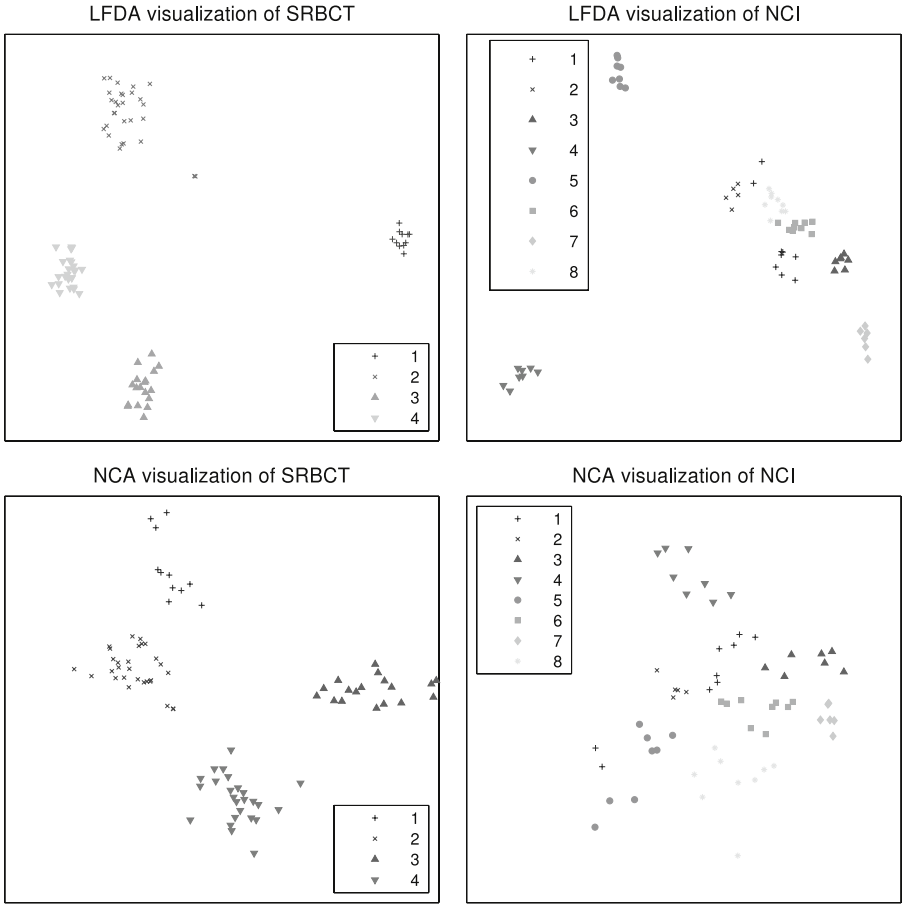


Fig. 9. Two-dimensional visualizations of the SRBCT data set (left column) and the NCI data (right column) obtained by LFDA and NCA. A detailed explanation can be found in the text.

the data set. We will present and exploit the obtained insights in a forthcoming study.

We demonstrate the advantages of LiRaM LVQ and its localized variant over LFDA and NCA: Fig. 12 shows the best visualizations we could achieve with this method. We varied the value k and l in the interval [1, 10] and for LFDA we achieved the best 1-NN error measures on the visualizations with $k = 6$ and $k = 9$ for 2D and 3D respectively. While certain classes (e.g. 14, alpine vegetation) seem to separate well, the overall discriminativity is limited. Only 50.4% accuracy can be achieved using a 1-NN classifier on the training data in the two-dimensional case and 89.6% in the three-dimensional case. For this particular data set the value of the parameter k has no significant influence on the quality of the LFDA-embedding of the training data. The computation of the 1-NN error on over three million data points of the test set was not practicable. Therefore we draw 100,000 points randomly from the test set and this reduced set serves as an approximation of the test-error. With the best LFDA we observed 61.3% and 93.75% 1-NN classification accuracy on the reduced test set for two and three dimensions, respectively. Table 1 shows the detailed comparison. The use of NCA turned out to be impractical due to excessive memory use. Therefore, we reduced the training set to 900 samples per class. We tried different values for the parameter l ranging from one to ten. The best results are shown in Fig. 12 (bottom row) for $k = 3$ and $k = 2$ in the 2D and 3D visualization respectively. On this data set the best NCA parametrization showed comparable or even better results than the LVQ approach. Nevertheless, some patience was necessary to get these results due to the computational complexity and the variation with respect to the parameter being huge. In the two-dimensional case the 1-NN accuracy ranged between 56.43% and 81.49% on the training set and in the 3-dim. case accuracies between 67.29% and 92.56% were observed. The other methods were shown to be faster and more robust with respect to the parametrization.

6. Summary and outlook

In this paper we present the LiRaM LVQ algorithm together with a localized variant, as a modification of Generalized Matrix LVQ (Schneider et al., 2009). It employs rectangular projection matrices to represent N-dim. feature vectors in an M-dim. space internally. This makes it possible to limit the rank of the relevance matrices used in GMLVQ which parameterize an adaptive distance measure. Obvious aims are to incorporate prior knowledge of the intrinsic dimension or to reduce the number of free parameters while maintaining good classification performance. In particular for high-dimensional data sets this can reduce the computational effort significantly. 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 LiRaM LVQ is an efficient method for determining discriminative, low-dimensional representations of labeled data and facilitates good generalization behavior. In LiRaM LVQ, the search for the appropriate subspace is guided directly by the classification performance in a single supervised training phase. This is in contrast to 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 LiRaM LVQ

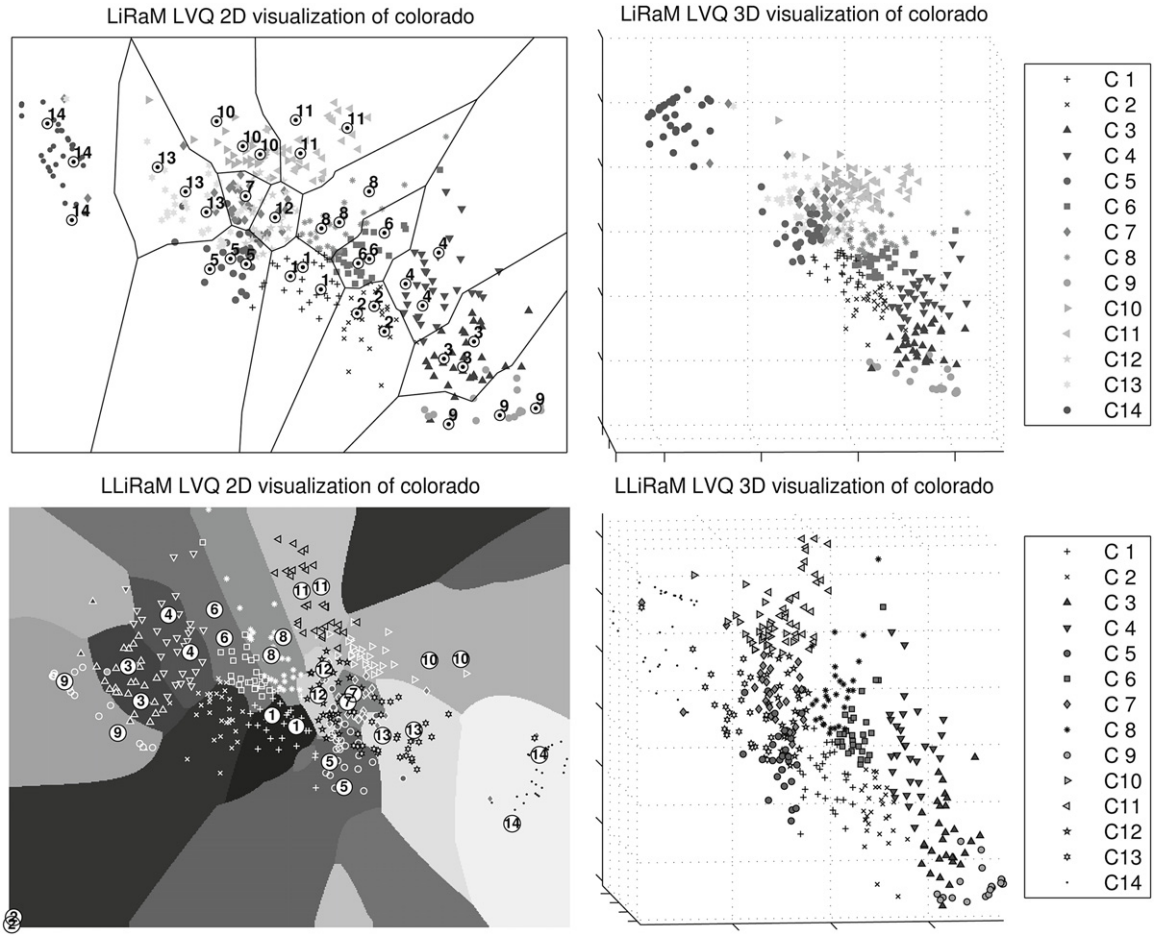


Fig. 10. Visualizations of a small subset of the Colorado data set acquired by the different methods.

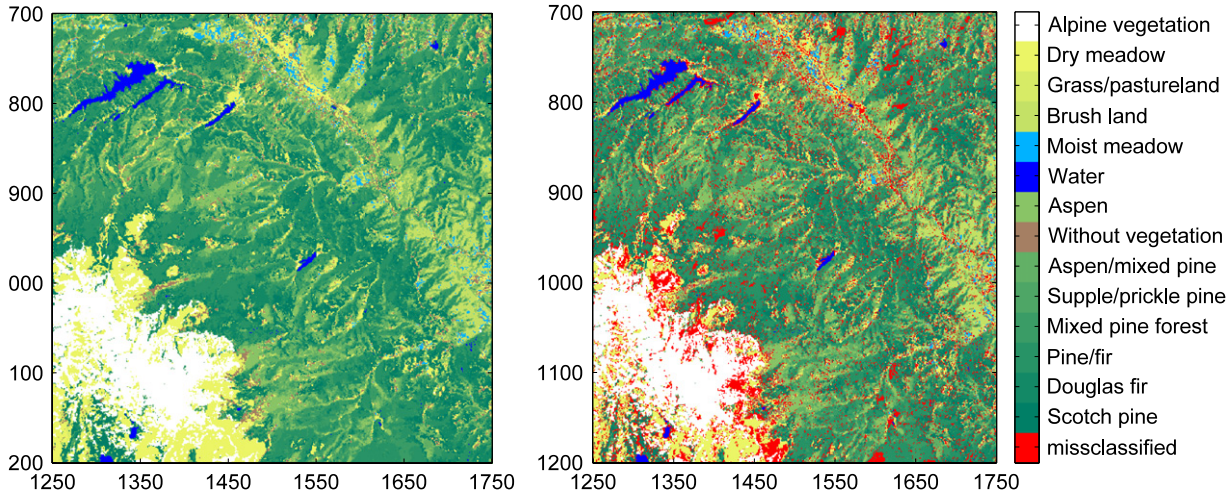


Fig. 11. The labels of a section of the Colorado satellite image (left panel) and the classification result obtained by the best run of LLiRaM LVQ in the 3D case (right panel). Detailed information about the class-wise accuracies can be found in the confusion matrix Table 3.

provides us with a discriminative visualization of the original data set. The algorithm results in linear or piece-wise linear decision boundaries dependent on the number of prototypes and classes. With the localized variant LLiRaM LVQ it is possible to visualize even more complicated non-linear decision boundaries. The key 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. Furthermore we compare the visualizations to some recent state-of-the-art supervised dimension reduction techniques, namely LFDA and NCA. The LFDA approach provides an analytical solution, but also depends on the computation of pairwise dissimilarities within samples of the same class. The results may differ a lot depending on the number k of neighbors used. For less complex data sets, like the four class SRCBT cancer data set, error free visualizations are possible. On other data sets LFDA showed worse results compared to the other methods. NCA showed good results

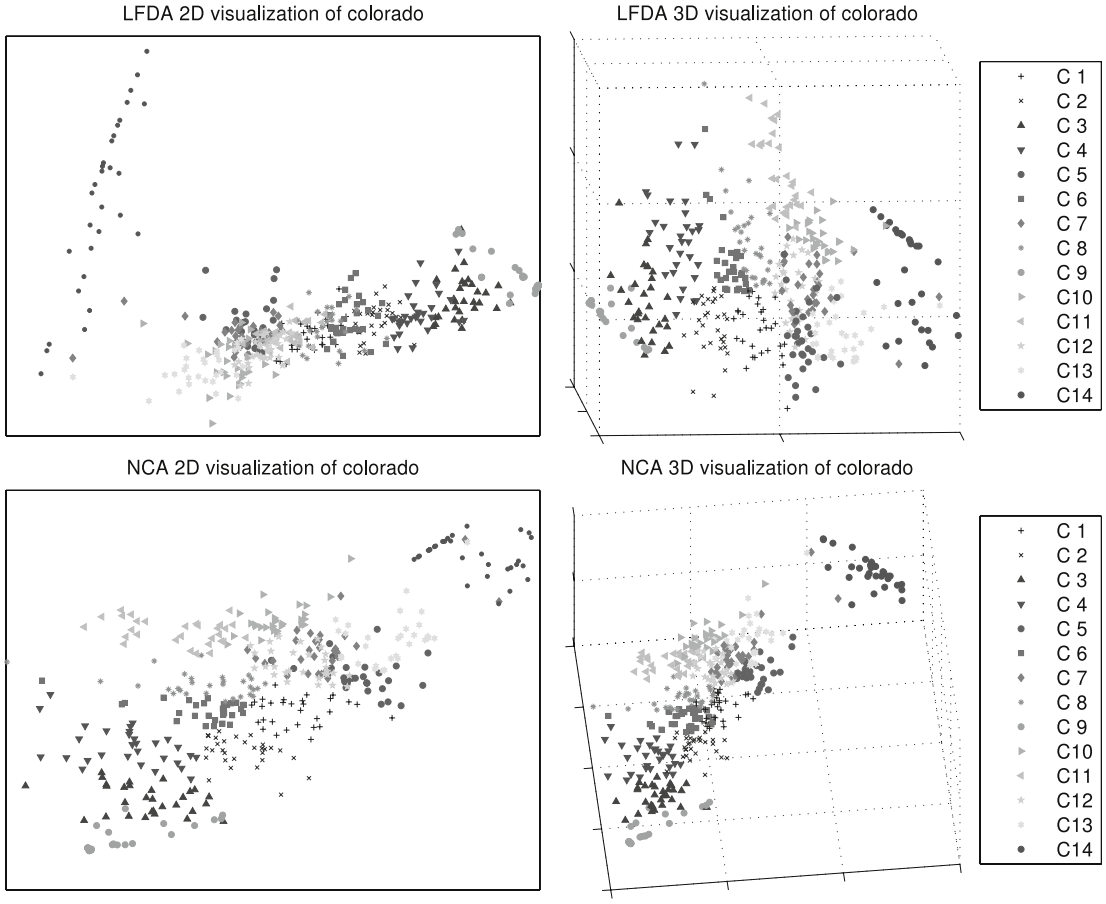


Fig. 12. Visualizations of a small subset of the Colorado data set acquired by the different methods.

Table 3
Confusion matrix of the 3D LLiRaM LVQ (Fig. 10 bottom right) on the Colorado data set.

C	Actual class															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	0	$\sum$
1	460 594	612	104	5	2 376	458	49	883	4	0	0	1 498	0	139	0	466 722
2	13 642	331 530	590	11 146	0	841	9	79	8	0	0	0	0	0	0	357 845
3	0	9 379	155 775	17 306	0	0	1	0	757	0	0	0	0	0	0	183 218
4	0	3 742	704	231 063	0	596	1	7	90	0	0	0	0	0	0	236 203
5	14 776	0	11	0	122 956	0	7 793	0	1	0	0	2 989	25 239	70	0	173 835
6	22 880	8 618	102	12 235	5	203 917	7	7 980	28	0	0	0	0	0	0	255 772
7	521	0	3	3	7 337	0	111 692	360	3	66	554	23 873	31 728	0	0	176 140
8	18 380	0	60	14	41	2 340	11	256 243	8	1	1 597	10 277	0	0	1	288 973
9	14	1 210	23 613	479	143	0	46	0	15 761	0	0	0	0	116	0	41 382
10	3	0	5	7	38	0	12 842	0	1	86 795	7 970	7 894	7 352	0	0	122 907
11	0	0	18	11	0	0	285	11 660	0	6 508	117 212	4 352	0	0	0	140 046
12	48 564	54	38	5	8 716	0	24 687	566	3	2 279	130	216 576	10 522	0	0	312 140
13	2 045	0	13	8	2 611	0	4 063	0	3	1 853	0	36	582 457	148	1	593 238
14	5	0	0	0	111	0	8 710	0	0	0	1	0	17 750	27 083	7	53 667
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\sum$	581 424	355 145	181 036	272 282	144 334	208 152	170 196	277 778	16 667	97 502	127 464	267 495	675 048	27 556	9	3402 088
Class-wise accuracy of the estimation in %																
	79.22	93.35	86.05	84.86	85.19	97.97	65.63	92.25	94.56	89.02	91.96	80.96	86.28	98.28	0	

in most cases. Its performance is also dependent on random initialization and the number of line searches l. NCA is based on the computation of pairwise dissimilarities which is expensive for large data sets. The LiRaM LVQ approach displays in all cases comparable or superior results on the investigated data sets. The computational effort depends on the target dimension, the number of prototypes and the number of samples for training. Unlike other methods, which require all pairwise dissimilarities, LiRaM LVQ computes distances of samples with respect to only a few prototypes. The observed influence of the number of protoytpes on the performance is relatively weak compared to the dependence on the neighborhood parameter in other methods.

The use of local or class-wise transformation matrices in LLiRaM LVQ allows for more complex decision boundaries. The decision boundary in the low-dimensional space is based on local matrices attached to the prototypes. Note that the dimension reduction itself is done in terms of a global linear projection. The concept of using local dissimilarities in combination with non-linear dimension reduction and visualization was recently discussed in Bunte et al. (2010b).

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, Section 5.4, we note that several features (intensities) essentially do not contribute to the highly discriminative linear combinations defined by Ω. 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 (Schneider et al., 2010). Most importantly, we plan to apply the LiRaM LVQ approach in various application domains, including the ones discussed above. An example application in the context of content based image retrieval is discussed in Bunte et al. (2011).

Acknowledgments

We would like to thank M. Augusteijn, Univ. of Colorado, for providing us with the satellite remote sensing data used in Section 5.5. We also thank M. Angelova, Northumbria Univ. Newcastle, for inspiring discussions and for drawing our attention to the gene expression data sets used in Section 5.4. This work was supported by the "Nederlandse Organisatie voor Wetenschappelijke Onderzoek (NWO)" under project code 612.066.620 and the German Research Foundation (DFG) in the project "Relevance Learning for Temporal Neural Maps" under project code HA2719/4-1.

References

Asuncion, A., Newman, D.J., Hettich, S., Blake, C.L., & Merz, C.J. 1998. UCI repository of machine learning databases. http://archive.ics.uci.edu/ml/.

Baudat, G., & Anouar, F. (2000). Generalized discriminant analysis using a kernel approach. *Neural Computation*, *12*, 2385–2404.

Bensmail, H., & Celeux, G. (1996). Regularized gaussian discriminant analysis through eigenvalue decomposition. *Journal of the American Statistical Association*, *91*, 1743–1748.

Biehl, M., Ghosh, A., & Hammer, B. (2007). Dynamics and generalization ability of LVQ algorithms. *Journal of Machine Learning Research*, *8*, 323–360.

Bojer, T., Hammer, B., Schunk, D., & von Toschanowitz, K.T. 2001. Relevance determination in learning vector quantization. In: Verleysen M. (Ed.) *Proc. of European symposium on artificial neural networks* (pp. 271–276).

Brand, M. 2003. Charting a manifold. Technical Report 15, Mitsubishi Electric Research Laboratories (MERL). http://www.merl.com/publications/TR2003-013/.

Bunte, K., Biehl, M., Jonkman, M. F., & Petkov, N. (2011). Learning effective color features for content based image retrieval in dermatology. *Pattern Recognition*, *44*(9), 1892–1902.

Bunte, K., Biehl, M., Petkov, N., & Jonkman, M.F. 2009. Adaptive metrics for content based image retrieval in dermatology. In: Verleysen M. (Ed.) *Proc. of European symposium on artificial neural networks* (pp. 129–134).

Bunte, K., Hammer, B., Schneider, P., & Biehl, M. 2009. Nonlinear discriminative data visualization. In: Verleysen M. (Ed.) *Proc. of European symposium on artificial neural networks* (pp. 65–70).

Bunte, K., Hammer, B., Wismüller, A., & Biehl, M. (2010a). Adaptive local dissimilarity measures for discriminative dimension reduction of labeled data. *Neurocomputing*, *73*(7–9), 1074–1092. Advances in Computational Intelligence and Learning—17th European Symposium on Artificial Neural Networks 2009.

Bunte, K., Hammer, B., Wismüller, A., & Biehl, M. (2010b). Adaptive local dissimilarity measures for discriminative dimension reduction of labeled data. *Neurocomputing*, *73*(7–9), 1074–1092. Advances in Computational Intelligence and Learning—17th European Symposium on Artificial Neural Networks 2009.

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern Classification*. John Wiley & Sons.

Eisen, M. B., Spellman, P. T., Brown, P. O., & Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *PNAS 95*, *25*, 14863–14868.

Ewing, R. M., & Cherry, J. M. (2001). Visualization of expression clusters using sammon's non-linear mapping. *Bioinformatics*, *17*, 658–659.

Faith, J., Mintram, R., & Angelova, M. (2006). Targeted projection pursuit for visualising gene expression data classifications. *Bioinformatics*, *22*, 2667–2673.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*, 179–188.

Friedman, J. H. (1989). Regularized gaussian discriminant analysis. *Journal of the American Statistical Association*, *84*, 165–175.

Fukunaga, K. (1990). *Computer Science and Scientific Computing Series*, *Introduction to Statistical Pattern Recognition* (2nd ed.) Academic Press.

Geng, X., Zhan, D.-C., & Zhou, Z.-H. (2005). Supervised nonlinear dimensionality reduction for visualization and classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, *35*(6), 1098–1107.

Goldberger, J., Roweis, S., Hinton, G., & Salakhutdinov, R. (2004). Neighbourhood components analysis. In *Advances in Neural Information Processing Systems 17* (pp. 513–520). MIT Press.

Hammer, B., Strickert, M., & Villmann, T. (2005a). On the generalization ability of GRLVQ networks. *Neural Processing Letters*, *21*(2), 109–120.

Hammer, B., Strickert, M., & Villmann, T. (2005b). Supervised neural gas with general similarity measure. *Neural Processing Letters*, *21*(1), 21–44.

Hammer, B., & Villmann, T. (2002). Generalized relevance learning vector quantization. *Neural Networks*, *15*(8–9), 1059–1068.

He, X., & Niyogi, P. (2003). Locality preserving projections. In *Advances in Neural Information Processing Systems 16*. MIT Press.

Iwata, T., Saito, K., Ueda, N., Stromsten, S., Griffiths, T. L., & Tenenbaum, J. B. (2007). Parametric embedding for class visualization. *Neural Computation*, *19*(9), 2536–2556.

Kaski, S., Sinkkonen, J., & Peltonen, J. (2001). Bankruptcy analysis with self-organizing maps in learning metrics. *IEEE Transactions on Neural Networks*, *12*, 936–947.

Khan, J., Wei, J. S., Ringnér, M., Saal, L. H., Ladanyi, M., & Westermann, F. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, *7*, 673–679.

Kohonen, T. (2001). *Self-Organizing Maps* (3rd ed.). Berlin, Heidelberg, New York: Springer.

Lee, E. K., Cook, D., Klinke, S., & Lumley, T. (2005). Projection pursuit for exploratory supervised classification. *Journal of Computational and Graphical Statistics*, *14*(4), 831–846.

Ma, B., Qu, H., & Wong, H. (2007). Kernel clustering-based discriminant analysis. *Pattern Recognition*, *40*(1), 324–327.

Memisevic, R., & Hinton, G. (2005). Multiple relational embedding. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in Neural Information Processing Systems 17* (pp. 913–920). Cambridge, MA: MIT Press.

Peltonen, J., Klami, A., & Kaski, S. (2004). Improved learning of riemannian metrics for exploratory analysis. *Neural Networks*, *17*, 1087–1100.

Sato, A. S., & Yamada, K. (1996). Generalized learning vector quantization. In D. S. Touretzky, M. Mozer, & M. E. Hasselmo (Eds.), *Advances in Neural Information Processing Systems*: *Vol. 8* (pp. 423–429). Cambridge, MA, USA: MIT Press.

Scherf, U., Ross, D. T., Waltham, M., et al. (2000). A gene expression database for the molecular pharmacology of cancer. *Nature Genetics*, *24*(3), 236–244.

Schleif, F.-M., Villmann, T., Hammer, B., Schneider, P., & Biehl, M. (2010). Generalized derivative based kernelized learning vector quantization. In C. Fyfe, P. Tiňo, D. Charles, C. García-Osorio, & H. Yin (Eds.), *Lecture Notes in Computer Science*: *Vol. 6283*. *IDEAL* (pp. 21–28). Springer.

Schneider, P., Biehl, M., & Hammer, B. (2009). Adaptive relevance matrices in learning vector quantization. *Neural Computation*, *21*(12), 3532–3561.

Schneider, P., Bunte, K., Hammer, B., & Biehl, M. (2010). Regularization in matrix relevance learning. *IEEE Transactions on Neural Networks*, *21*(5), 831–840.

Song, L., Smola, A. J., Borgwardt, K. M., & Gretton, A. (2008). Colored maximum variance unfolding. In J. C. Platt, D. Koller, Y. Singer, & S. T. Roweis (Eds.), *NIPS*. MIT Press.

Sugiyama, M., & Roweis, S. (2007). Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *Journal of Machine Learning Research*, *8*, 1027–1061.

Tenenbaum, J. B., Silva, V. d., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*(5500), 2319–2323.

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, *9*, 2579–2605.

Venna, J., Peltonen, J., Nybo, K., Aidos, H., & Kaski, S. (2010). Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research*, *11*, 451–490.

Villmann, T., Merenyi, E., & Hammer, B. (2003). Neural maps in remote sensing image analysis. *Neural Networks*, *16*(3–4), 389–403.

Weinberger, K. Q., Blitzer, J., & Saul, L. K. (2006). Distance metric learning for large margin nearest neighbor classification. *Advances in Neural Information Processing Systems*, *18*, 1473–1480.

Zhang, L., Zhang, A., & Ramanathan, M. (2004). Vizstruct: exploratory visualization for gene expression profiling. *Bioinformatics*, *20*, 85–92.