Adaptive Local Dissimilarity Measures for Discriminative Dimension Reduction of Labeled Data

Kerstin Bunte[a], Barbara Hammer[c], Axel Wismüller[b], Michael Biehl[a]

[a] *University of Groningen - Institute of Mathematics and Computing Sciences*
P.O. Box 407, 9700 AK Groningen - The Netherlands
[b] *Depts. of Radiology and Biomedical Engineering, University of Rochester, New York*
601 Elmwood Avenue, Rochester, NY 14642-8648, U.S.A.
[c] *Clausthal University of Technology - Institute of Informatics*
Julius Albert Strasse 4, 38678 Clausthal-Zellerfeld - Germany

Abstract

Due to the tremendous increase of electronic information with respect to the size of data sets as well as their dimension, dimension reduction and visualization of high-dimensional data has become one of the key problems of data mining. Since embedding in lower dimensions necessarily includes a loss of information, methods to explicitly control the information kept by a specific dimension reduction technique are highly desirable. The incorporation of supervised class information constitutes an important specific case. The aim is to preserve and potentially enhance the discrimination of classes in lower dimensions. In this contribution we use an extension of prototype-based local distance learning, which results in a nonlinear discriminative dissimilarity measure for a given labeled data manifold. The learned local distance measure can be used as basis for other unsupervised dimension reduction techniques, which take into account neighborhood information. We show the combination of different dimension reduction techniques with a discriminative similarity measure learned by an extension of Learning Vector Quantization (LVQ) and their behavior with different parameter settings. The methods are introduced and discussed in terms of artificial and real world data sets.

Key words: Dimension Reduction, Learning Vector Quantization, Visualization

1. Introduction

The amount of electronic data doubles roughly every 20 months[1], and its sheer size makes it impossible for humans to manually scan through the available information. At the same time, rapid technological developments cause an increase of data dimension, e.g. due to the increased sensitivity of sensor technology (such as mass spectrometry) or the improved resolution of imaging techniques. This causes the need for reliable dimension reduction and data visualization techniques to allow humans to rapidly inspect large portions of data using their impressive and highly sensitive visual perception capabilities.

Dimension reduction and visualization constitutes an active field of research, see e.g. [2, 3, 4] for recent overviews. The embedding of high-dimensional data into lower dimension is necessarily linked to loss of information. In the last decades an enormous number of unsupervised dimension reduction methods has been proposed. In general, unsupervised dimension reduction is an ill-posed problem since a clear specification which properties of the data should be preserved, is missing. Standard criteria, for instance the distance measure employed for neighborhood assignment, may turn out unsuitable for a given data set, and relevant information often depends on the situation at hand.

If data labeling is available, the aim of dimension reduction can be defined clearly: the preservation of the classification accuracy in a reduced feature space. Supervised linear dimension reducers are for example the Generalized Matrix Learning Vector Quantization (GMLVQ) [5], Linear Discriminant Analysis (LDA) [6], targeted projection pursuit [7], and discriminative component analysis [8]. Often, however, the classes cannot be separated by a linear classifier while a nonlinear data projection better preserves the relevant information. Examples for nonlinear discriminative visualization techniques include, extensions of the Self Organizing Map (SOM) incorporating class labels [9] or more general auxiliary information [10]. In both cases, the metric of SOM is adjusted such that it emphasizes the given auxiliary information and, consequently, SOM displays the aspects relevant for the given labeling. Further supervised dimension reduction techniques are model-based visualization [11] and parametric embedding [12]. In addition, linear schemes such as LDA can be kernelized yielding a nonlinear supervised dimension reduction scheme [13]. These models have the drawback that they are often very costly (squared or cubic with respect to the number of data points). Recent

approaches provide scalable alternatives, sometimes at the cost of non convexity of the problem [14, 15, 16]. However, the kernel has to be chosen prior to training and no metric adaptation according to the given label information takes place.

The aim of this paper is to identify and investigate principled possibilities to combine an adaptive metric and recent visualization techniques towards a discriminative approach. We will exploit the discriminative scheme exemplary for different types of visualization, necessarily restricting the number of possible combinations to exemplary cases. A number of alternative combinations of metric learning and data visualization as well as principled alternatives to arrive at discriminative visualization techniques (such as e.g. colored Maximum Variance Unfolding [17]) will not be tackled in this paper.

In this contribution we combine prototype-based matrix learning schemes, which result in local discriminative dissimilarity measures and local linear projections of the data, with different neighborhood based nonlinear dimension reduction techniques and a charting technique. The complexity of the matrix learning technique is only linear in the number of points, their dimension and can be controlled by the number of the prototypes and sweep through the training set (t), leading to an overall algorithm complexity of only $\mathcal{O}(S \cdot N \cdot m \cdot t)$. In the second step unsupervised techniques like manifold charting [18], Isomap [19], Locally Linear Embedding (LLE) [20], the Exploration Observation Machine (XOM) [21] and Stochastic Neighbor Embedding (SNE) [22] are performed incorporating the supervised information from the LVQ approach. This leads to supervised nonlinear dimension reduction and visualization techniques. Note that in contrast to the matrix learning scheme, which computes only the neighborhood of the data points to normally much less prototypes, the combination with another dimension reduction technique may need the computation of the distances of all data points. This is at least a quadratic problem but can be moderated by approximations [23, 24, 25].

The following section gives a short overview over the techniques. We focus on the question in how far local linear discriminative data transformations as provided by GMLVQ offer principled possibilities to extend standard unsupervised visualization tools to discriminative visualization. Section 3 discusses the different approaches for one artificial and three real world data sets and compares the results to popular supervised as well as unsupervised dimension reduction techniques. Finally we conclude in section 4.

2. Supervised Nonlinear Dimension Reduction

For general data sets a global linear reduction to lower dimensions may not be sufficient to preserve the information relevant for classification. In [3] it is argued that the combination of several local linear projections to a nonlinear mapping can yield promising results. We use this concept and learn discriminative local linear low-dimensional projections from labeled data using an efficient prototype based learning scheme, Generalized Matrix Learning Vector Quantization (GMLVQ). Locally linear projections which result from this first step provide, on the one hand, local transformations of the data points which preserve the information relevant for the classification as much as possible. Instead of the local coordinates, local distances induced by these local representation of data can be considered. As a consequence, visualization techniques which rely on local coordinate systems or local distances, respectively, can be combined with this first step to arrive at a discriminative global nonlinear projection method. This way, an incorporation into techniques such as manifold charting [18], Isomap [19], Locally Linear Embedding (LLE) [20], stochastic neighbor embedding (SNE) [22], and the Exploration Observation Machine (XOM) [21] becomes possible.

The following subsections give a short overview over the initial prototype based matrix learning scheme and the different visualization algorithms.

2.1. Localized LiRaM LVQ

Learning vector quantization (LVQ) [26] constitutes a particularly intuitive classification algorithm which represents data by means of prototypes. LVQ itself constitutes a heuristic algorithm, hence extensions have been proposed for which convergence and learnability can be guaranteed [27, 28]. One particularly crucial aspect of LVQ schemes is the dependency on the underlying metric, usually the Euclidean metric, which may not suit the underlying data structure. Therefore, general metric adaptation has been introduced into LVQ schemes [28, 29]. Recent extensions parameterize the distance measure in terms of a relevance matrix, the rank of which may be controlled explicitly. The algorithm suggested in [5] can be employed for linear dimension reduction and visualization of labeled data. The local linear version presented here provides the ability to learn local low-dimensional projections and combine them into a nonlinear global embedding using charting techniques or projection methods based on local data topologies or local distances. Several schemes for adaptive distance learning exist, for example

Large Margin Nearest Neighbor (LMNN)[30] to name just one. We compared the LMNN technique with the LVQ based approach on the basis of a Content Based Image Retrieval application in an earlier publication (see [31]). Furthermore it should be mentioned that LMNN learns a global distance measure and it is not obvious how to adapt the algorithm to perform more powerful local distance learning like we propose in this contribution.

We consider training data $\boldsymbol{x}_i \in \mathbb{R}^N$, $i = 1 \ldots S$ with labels y_i corresponding to one of C classes respectively. The aim of LVQ is to find m prototypes $\boldsymbol{w}_j \in \mathbb{R}^N$ with class labels $c(\boldsymbol{w}_j) \in \{1, \ldots, C\}$ such that they represent the classification as accurately as possible. A data point $\boldsymbol{x}_i$ is assigned to the class of its closest prototype $\boldsymbol{w}_j$ where $d(\boldsymbol{x}_i, \boldsymbol{w}_j) \leq d(\boldsymbol{x}_i, \boldsymbol{w}_l)$ for all $j \neq l$. d usually denotes the squared Euclidean distance $d(\boldsymbol{x}_i, \boldsymbol{w}_j) = (\boldsymbol{x}_i - \boldsymbol{w}_j)^\top (\boldsymbol{x}_i - \boldsymbol{w}_j)$. Generalized LVQ (GLVQ) [32] adapts prototype locations by minimizing the cost function

$$E_{\mathrm{GLVQ}} = \sum_{i=1}^{S} \Phi \left(\frac{d(\boldsymbol{w}_J, \boldsymbol{x}_i) - d(\boldsymbol{w}_K, \boldsymbol{x}_i)}{d(\boldsymbol{w}_J, \boldsymbol{x}_i) + d(\boldsymbol{w}_K, \boldsymbol{x}_i)} \right) \quad , \tag{1}$$

where $\boldsymbol{w}_J$ denotes the closest prototype with the same class label as $\boldsymbol{x}_i$, and $\boldsymbol{w}_K$ is the closest prototype with a different class label. Φ is a monotonic function, e.g. the logistic function or the identity. In this work we use the identity. This cost function aims at an adaptation of the prototypes such that a large hypothesis margin is obtained, this way achieving correct classification and, at the same time, robustness of the classification, see [33]. A learning algorithm can be derived from the cost function E_{GLVQ} by means of a stochastic gradient descent as shown in [28, 27].

Matrix learning in GLVQ (GMLVQ) [29, 33] substitutes the usual squared Euclidean distance d by a more advanced dissimilarity measure which contains adaptive parameters, thus resulting in a more complex and better adaptable classifier. In [5], it was proposed to choose the dissimilarity as

$$d^{\Lambda_j}(\boldsymbol{w}_j, \boldsymbol{x}_i) = (\boldsymbol{x}_i - \boldsymbol{w}_j)^\top \Lambda_j (\boldsymbol{x}_i - \boldsymbol{w}_j) \tag{2}$$

with an adaptive, symmetric and positive semi-definite matrix $\Lambda_j \in \mathbb{R}^{N \times N}$ locally attached to each prototype $\boldsymbol{w}_j$. The dissimilarity measure Eq. (2) possesses the shape of a Mahalanobis distance. Note, however, that the precise matrix is determined in a discriminative way according to the given labeling, such that severe differences from the standard Mahalanobis distance based on correlations can be observed. By setting $\Lambda_j = \Omega_j^\top \Omega_j$ semi-definiteness and

symmetry is guaranteed. Optimization takes place by a stochastic gradient descent of the cost function E_{GLVQ} in Eq. (1), with the distance measure d substituted by d^{Λ_j} (see Eq. (2)). After each training epoch (sweep through the training set) the matrices are normalized to $\sum_i [\Lambda_j]_{ii} = 1$ in order to prevent degeneration to 0. An additional regularization term in the cost function proportional to $-\ln(\det(\Omega_j \Omega_j^\top))$ can be used to enforce full rank M of the relevance matrices and prevent over-simplification effects, see [34].

The cost function of GMLVQ is non convex and, in consequence, different local optima can occur which lead to different subsequent data visualizations. The non convexity of the cost function is mainly due to the discrete data assignments to prototypes which is not unique in particular for realistic data sets with overlapping classes. In the experiments, different assignments and, in consequence, different visualizations could be observed, where these visualizations focus on different relevant facets of the given data sets.

The choice $\Omega_j \in \mathbb{R}^{M \times N}$ with $M \leq N$ transforms the data locally to an M-dimensional feature space. It can be shown that the adaptive distance $d^{\Lambda_j}(\boldsymbol{w}_j, \boldsymbol{x}_i)$ in Eq. (2) equals the squared Euclidean distance in the transformed space under the transformation $\boldsymbol{x} \mapsto \Omega_j \boldsymbol{x}$, because $d^{\Lambda_j}(\boldsymbol{w}_j, \boldsymbol{x}_i) = [\Omega_j \boldsymbol{x}_i - \Omega_j \boldsymbol{w}_j)]^2$. The target dimension M must be chosen in advance by intrinsic dimension estimation or according to available prior knowledge. For visualization purposes, usually a value of two or three is appropriate. At the end of the learning process the algorithm provides a set of prototypes $\boldsymbol{w}_j$, their labels $c(\boldsymbol{w}_j)$, and corresponding projections Ω_j and distances d^{Λ_j}. For every prototype, a low dimensional embedding $\boldsymbol{\xi}_i$ of each data point $\boldsymbol{x}_i$ is then given by

$$P_j(\boldsymbol{x}_i) = \Omega_j \boldsymbol{x}_i = \boldsymbol{\xi}_i \tag{3}$$

This projection is a meaningful discriminative projection in the neighborhood of a prototype, i.e for a data point $\boldsymbol{x}_i$, usually the projection Ω_j of its closest prototype $\boldsymbol{w}_j$ is considered. This way, a global mapping is given as

$$\boldsymbol{x}_i \mapsto P_j(\boldsymbol{x}_i) = \Omega_j \boldsymbol{x}_i \text{ with } d^{\Lambda_j}(\boldsymbol{w}_j, \boldsymbol{x}_i) = \min_k d^{\Lambda_k}(\boldsymbol{w}_k, \boldsymbol{x}_i). \tag{4}$$

We will refer to this prototype and matrix learning algorithm as Limited Rank Matrix LVQ (LiRaM LVQ), and we will address the local linear mappings induced by LiRaM LVQ as LiRaM LVQ mappings. Note that the effort to obtain these projections depends linearly on the number of data, the number of prototypes, the number of training epochs, and the dimensionalities N and M of the matrices.

Note that Ω_j is not uniquely given by the cost function Eq. (1) and it varies for different initializations, because the dissimilarity is invariant under operations such as rotation of the matrices. If a unique representation $\widehat{\Omega}_j$ of Ω_j is needed for comparison, the unique square root of Λ_j is chosen [35].

LiRaM LVQ directly provides a global linear discriminative embedding of data if only one global adaptive matrix $\Omega = \Omega_j$ is used, as demonstrated in [5]. Alternatively, one can consider the local data projections provided by the LiRaM LVQ mappings on the receptive fields of the prototypes as defined in Eq. (4). However, the cost function together with the distance definition does not ensure that these local projections align correctly and that they do not overlap when shown in one coordinate system. Rather, the projections provide widely unconnected mappings to low dimensions which offer only a locally valid visualization. Nevertheless the mapping defined by Eq. (4) can give a first intuition about the problematic samples and distinguish "easy" classes from more difficult ones. Therefore, we will use this projection as a comparison in the experiments.

In order to achieve interpretable global nonlinear mappings of the data points we have to align the local information provided by the local projections. This can be done in different ways, using an explicit charting technique of the maps or using visualization techniques based on the local distances provided by this method. In the following, we introduce a few principled possibilities to combine the information of LiRaM LVQ and unsupervised visualization techniques to achieve a global nonlinear discriminative visualization.

2.2. Information provided by LiRaM LVQ for discriminative visualization

LiRaM LVQ determines prototypes and matrices based on a discriminative classification task. These parameters induce important discriminative information which can be plugged into different visualization schemes.

Local coordinates

As already stated, LiRaM LVQ gives rise to local linear projection maps P_j as defined in Eq. (3) which assign local projection coordinates to every data point $\boldsymbol{x}_i$. These projections can be accompanied by values which indicate the responsibility r_{ji} of mapping j for data point i. Crisp responsibilities are obtained by means of the receptive fields, setting r_{ji} to 1 iff $\boldsymbol{w}_j$ is the winner for $\boldsymbol{x}_i$. Alternatively, soft assignments can be obtained by centering Gaussian curves of appropriate bandwidth at the prototypes.

These two ingredients constitute a sufficient input for data visualization methods which rely on local linear projections of the data only, such as manifold charting, Locally Linear Coordination (LLC) [3] and Local Tangent Space Alignment (LTSA) [36]. Basically, those methods arrive at a global embedding of data based on local coordinates by gluing the points together such that the overall mapping is consistent with the original data points as much as possible. The methods differ in the precise cost function which is optimized: Manifold charting relying on the sum squared error of points at overlapping pieces of the local charts, while LLC focuses on the local topology and tries to minimize the reconstruction error of points from their neighborhood. Both approaches provide explicit maps of the data manifold to low dimensions, such that Out-of-Sample extensions are immediate. As an example for this principle, we will investigate the combination of local linear maps and manifold charting.

Global distances

The LiRaM LVQ learning procedure provides discriminative local distances induced by the matrices Λ_j in the receptive field of prototype $\boldsymbol{w}_j$. We use this observation to define a discriminative dissimilarity measure for the given data points. We define the dissimilarity of a point $\boldsymbol{x}_i$ to a point $\boldsymbol{x}$:

$$d(\boldsymbol{x}_i, \boldsymbol{x}) = (\boldsymbol{x}_i - \boldsymbol{x})^\top \Lambda_j (\boldsymbol{x}_i - \boldsymbol{x}) \text{ where } d^{\Lambda_j}(\boldsymbol{x}_i, \boldsymbol{w}_j) = \min d^{\Lambda_k}(\boldsymbol{x}_i, \boldsymbol{w}_k) \quad (5)$$

using the distance measure Λ_j induced by the closest prototype $\boldsymbol{w}_j$ of $\boldsymbol{x}_i$. Note that this definition leads to asymmetric dissimilarities, where $d(\boldsymbol{x}_i, \boldsymbol{x}_j) \neq d(\boldsymbol{x}_j, \boldsymbol{x}_i)$ can hold. It is block wise symmetric for data samples with the same winner prototype in the classification task. Further, due to the nature of the LiRaM LVQ cost function, the dissimilarity measure constitutes a valid choice only within or near receptive fields. The dissimilarity of far away points which are not located in the same or proximous receptive fields can be seen only as a rough estimation of a valid dissimilarity.

The global dissimilarities defined by Eq. (5) can be used directly within visualization schemes which are based on distance preservation. If necessary, the dissimilarity matrix can be symmetrized prior to the mapping. Distance based visualization methods include classical multidimensional scaling (MDS), Sammon's map, Stochastic Neighbor Embedding (SNE), t-Distributed SNE (t-SNE), and the Exploration Observation Machine (XOM), to name just a few [3, 22, 37, 21]. It can be expected that the combination of the

global discriminative dissimilarities as given by Eq. (5) yields to an appropriate visualization of the data only if the visualization method mainly focuses on the close points, since the dissimilarity of far away points can only be seen as a guess in this case. Thus, classical MDS is likely to fail, while SNE or XOM seem more promising due to their focus on local distances. Further, these methods usually provide an embedding of the points only without giving an explicit map, such that Out-of-Sample extensions are not immediate. As an example, we will investigate the combination of the global dissimilarity matrix with SNE and XOM, respectively, in the following.

In contrast to the charting approach, the ranks M of the distance matrices Λ_j can be chosen larger than the embedding dimension in these cases, using e.g. full ranks or the intrinsic dimension of the data manifold.

Local distances or neighborhood

The problem that the dissimilarity measure as defined in Eq. (5) should preferably only be used to compare data within a receptive field or in neighbored receptive fields is avoided by visualization techniques which explicitly rely on local distances only. Instances of such visualization techniques are given by Isomap, Laplacian Eigenmaps, Locally Linear Embedding (LLE)[3] and Maximum Variance Unfolding (MVU)[38]. These methods use the local neighborhood of a data point, i.e. its k nearest neighbors (k-NN neighborhood) or the points in an ϵ-ball (ϵ-neighborhood), and try to preserve properties of these neighborhoods. Obviously, local neighborhoods can readily be computed based on the dissimilarities given by Eq. (5), thus a discriminative extensions of these methods is offered this way.

Isomap extends local distances within the local neighborhoods to a global measure by means of the graph distance, using simple MDS after this step. Laplacian Eigenmaps use the neighborhood graph and try to map data points such that close points remain close in the projection. LLE also relies on the local neighborhood, but it tries to preserve the local angles of points rather than the distances. Obviously, these methods can be transferred to discriminative visualization techniques by using the local neighborhood as given be the local discriminative distances and, if required, the local discriminative distances themselves. As an example, we will investigate the combination of Isomap and LLE with this discriminative technique. Again, these techniques provide a map of the given data points only, rather than an explicit embedding function.

Now we introduce four exemplary discriminative projection methods, cov-

ering the different possibilities to combine the information given by LiRaM LVQ and visualization techniques. We will compare these methods to a naive embedding directly given by the local linear maps as a baseline, Linear Discriminant Analysis (LDA) [6] (if applicable) as a classical linear discriminative visualization tool, and t-SNE as one of the currently most powerful unsupervised visualization techniques. Further, we will emphasize the effect of discriminative information by presenting the result of the corresponding unsupervised projection method.

2.3. Combination of Local Linear Patches by Charting

The charting technique introduced in [18] provides a frame for unsupervised dimension reduction by decomposing the sample data into locally linear patches and combining them into a single low-dimensional coordinate system. This procedure can be turned into a discriminative visualization scheme by using the M low-dimensional local linear projections $P_j(\boldsymbol{x}_i) \in \mathbb{R}^M$ for every data point $\boldsymbol{x}_i$ and every local projection Ω_j provided by localized LiRaM LVQ in the first step. The second step of the charting method can then directly be used to combine these maps. In charting, the local projections $P_j(\boldsymbol{x}_i)$ are weighted by their responsibilities r_{ji} which quantify the overlap of neighbored charts. Here we choose responsibilities induced by Gaussians centered at the prototypes, since a certain degree of overlap is needed for a meaningful charting step:

$$r_{ji} \propto \exp(-(\boldsymbol{x}_i - \boldsymbol{w}_j)^\top \Lambda_j (\boldsymbol{x}_i - \boldsymbol{w}_j)/\sigma_j) \ , \tag{6}$$

where $\sigma_j > 0$ constitutes an appropriate bandwidth. Further, we have to normalize these responsibilities $\sum_j r_{ji} = 1$ in order to apply charting. Since the combination step needs a reasonable overlap of neighbored patches, the bandwidth σ_j must be chosen to ensure this property. We set σ_j to a fraction α ($0 < \alpha < 1$) of the mean distance to the k nearest prototypes in the original feature space

$$\sigma_j = \alpha \cdot \frac{1}{k} \sqrt{\sum_{\boldsymbol{w}_l \in \mathcal{N}_k(\boldsymbol{w}_j)} d^{\Lambda_j}(\boldsymbol{w}_j, \boldsymbol{w}_l)} \tag{7}$$

where $\mathcal{N}_k(\boldsymbol{w}_j)$ denotes the k closest prototypes to $\boldsymbol{w}_j$.

Manifold charting minimizes a convex cost function that measures the amount of disagreement between the linear models on the global coordinates of the data points. The charting technique finds affine mappings A_j from the

data representations $P_j(\boldsymbol{x}_i)$ to the global coordinates that minimize the cost function

$$E_{\text{charting}} = \sum_{i=1}^{S} \sum_{j=1}^{m} \sum_{k=1}^{m} r_{ji} r_{ki} ||A_j(P_j(\boldsymbol{x}_i)) - A_j(P_k(\boldsymbol{x}_i))||^2 \ . \tag{8}$$

This function is based on the idea that whenever two linear models possess a high responsibility for a data point, the models should agree on the final coordinates of that point. The cost function can be rewritten as a generalized eigenvalue problem and an analytical solution can be found in closed form. The projection is given by the mapping $\boldsymbol{x}_i \mapsto \boldsymbol{\xi}_i = \sum_j r_{ji} A_j(P_j(\boldsymbol{x}_i))$. We refer to [18] for further details. Interestingly, an explicit map of the data manifold to low dimensions is obtained this way. Further, the charting step is linear in the number of data points S. we refer to the extension of charting by local discriminative projections as charting$^+$ in the following.

2.4. Discriminative Locally Linear Embedding

Locally linear embedding (LLE) [20] uses the criterion of topology preservation for dimension reduction. The idea is to reconstruct each point $\boldsymbol{x}_i$ by a linear combination of its nearest neighbors with coefficients W_i and to project data points such that this local representation of the data is preserved as much as possible.

The first step of the LLE algorithm is the determination of neighbors $\mathcal{N}_i$ for each data point $\boldsymbol{x}_i$. Typical choices are either the k closest points or all points lying inside an ϵ-ball with center $\boldsymbol{x}_i$. Following the ideas of supervised LLE [39] and probability-based LLE [40] we take the label information into account by using the distance measure defined in Eq. (5) to determine the k nearest neighbors of each point. The neighborhood of $\boldsymbol{x}_i$ is referred to as $\mathcal{N}_i$.

The second and third step of the LLE approach remain unchanged. In step two the reconstruction weights W for every data point are computed by minimizing the squared reconstruction error

$$E_{\text{LLE}}(W) = \sum_{i=1}^{S} ||\boldsymbol{x}_i - \sum_{j \in \mathcal{N}_i} w_{ij} \boldsymbol{x}_j||^2 \ , \tag{9}$$

by constrained linear fits such that $\sum_j w_{ij} = 1$. The entries w_{ij} of the matrix W weight the neighbors for the reconstruction of $\boldsymbol{x}_i$. Due to the constraints, this scheme is invariant to rotations, rescalings and translations of the data

points. The third step of LLE bases on the idea, that these geometric properties should also be valid for a low-dimensional representation of the data. Thus, the low-dimensional representations $\boldsymbol{\xi}$ are found by minimizing the cost function

$$\mathrm{Cost}_{\mathrm{LLE}}(\boldsymbol{\xi}) = \sum_i \left(\boldsymbol{\xi}_i - \sum_{j \in \mathcal{N}_i} w_{ij} \boldsymbol{\xi}_j \right)^2 = \mathrm{tr}(\boldsymbol{\xi}^\top (I - W)^\top (I - W) \boldsymbol{\xi}) \ , \quad (10)$$

with I being the identity matrix. The smallest D nonzero eigenvalues of $(I - W)^\top (I - W)$ yield the final embedding $\boldsymbol{\xi}$. Further details can be found in [20]. Advantages of this method are the elegant theoretical foundation which allows an analytical solution. LLE requires the solution of an S-by-S eigenproblem with S being the number of data points, i.e. the effort is squared in S. As reported in [41], the parameters must be tuned carefully. The method provides a mapping of the given points only rather than an explicit embedding function. We refer to this discriminative extension of LLE by LLE$^+$ in the following.

2.5. Discriminative Isomap

Isomap [19] is an extension of metric Multi-Dimensional Scaling (MDS) which uses distance preservation as criterion for the dimension reduction. It performs a low dimensional embedding based on the pairwise distance between data points. Whereas metric MDS is based on the Euclidean metric for all pairwise distances, Isomap incorporates the so called graph distances as an approximation of the geodesic distances. For this purpose, a weighted neighborhood graph is constructed by connecting points i and j if their distance is smaller than ϵ (ϵ-Isomap), or if j is one of the k nearest neighbors of i (k-Isomap). Global distances between points are computed using shortest paths in this neighborhood graph. The local neighborhood graph can serve as an interface to incorporate discriminative information provided by LiRaM LVQ. We use the distances defined by Eq. (5) to determine the k nearest neighbors and to weight the edges in the neighborhood graph. Afterwards, we simply apply the same projection technique as original Isomap.

The Isomap algorithm shares the same model type and optimization procedure as Principal Component Analysis (PCA) and metric MDS, thus a global optimum can be determined analytically. While PCA and MDS are designed for linear submanifolds, Isomap can handle developable manifolds due to its use of geodesic distances. However, it may yield disappointing

results if applied to not developable manifolds. Furthermore, the quality of the approximation of the geodesic distances by the graph distances may be sensitive to the choice of the parameters. A similar problem exists for virtually all local manifold estimation techniques which rely on critical parameters to define the local neighborhood such as the number of neighbors or the neighborhood size. For details we refer to [19]. Note that this method is of cubic complexity with respect to the number of data because of the necessity to compute all shortest paths within the neighborhood graph with S vertices. As before, no explicit map is obtained for the embedding when using Isomap. We refer to this discriminative extension of Isomap as Isomap$^+$ in the following.

2.5.1. Discriminative Stochastic Neighbor Embedding

Stochastic Neighbor Embedding (SNE) constitutes an unsupervised projection which follows a probability based approach. A Gaussian function is centered at every data point $\boldsymbol{x}_i$ which induces a conditional probability of a point $\boldsymbol{x}_j$ given $\boldsymbol{x}_i$

$$p_{j|i} = \frac{\exp\left(-d(\boldsymbol{x}_i, \boldsymbol{x}_j)/2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-d(\boldsymbol{x}_i, \boldsymbol{x}_k)/2\sigma_i^2\right)} \ , \tag{11}$$

where σ_i is the variance and d denotes the dissimilarity measure, which is chosen as squared Euclidean distance in the original approach. Projection takes place in such a way that these conditional probabilities are preserved as much as possible for the projected points in the low-dimensional space. More precisely, for counterparts $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$ of the data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ the conditional probability is defined similar to Eq. (11)

$$q_{j|i} = \frac{\exp\left(-||\boldsymbol{\xi}_i, \boldsymbol{\xi}_j||^2\right)}{\sum_{k \neq i} \exp\left(-||\boldsymbol{\xi}_i, \boldsymbol{\xi}_k)||^2\right)} \tag{12}$$

with fixed bandwidth $\frac{1}{\sqrt{2}}$. The goal of SNE is to find a low-dimensional data representation that minimizes the mismatch between $p_{j|i}$ and $q_{j|i}$. This is done by the minimization of the sum of the Kullback-Leibler divergences

$$E_{\mathrm{SNE}} = \sum_i \sum_j p_{j|i} \log \frac{p_{j,i}}{q_{j|i}} \ . \tag{13}$$

The minimization of this objective function is difficult and may stuck in local minima. Details can be found in [22].

An important parameter of SNE is the so-called perplexity which is used to determine the bandwidths of the Gaussians in the data space based on the effective number of local neighbors. The performance of SNE is fairly robust to changes in the perplexity and typical values are between 5 for small data sets and 50 for large datasets with more than 10 000 instances.

It is easily possible to incorporate discriminative information into SNE by choosing the distances $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ in Eq. (11) as discriminative distances as provided by Eq. (5). Then, the subsequent steps can be done in the same way as in standard SNE. This way, a discriminative embedding of data which displays quadratic effort in S can be obtained.

2.6. Discriminative Exploration Observation Machine (XOM)

The Exploratory Observation Machine (XOM) has recently been introduced as a novel computational framework for structure-preserving dimension reduction [42, 43]. It has been shown that XOM can simultaneously contribute to several different domains of advanced machine learning, scientific data analysis, and visualization, such as non-linear dimension reduction, data clustering, pattern matching, constrained incremental learning, and the analysis of non-metric dissimilarity data [44, 45].

The XOM algorithm can be resolved into three simple, geometrically intuitive steps.

Step 1: Define the topology of the input data in the high-dimensional data space by computing distances $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ between the data vectors $\boldsymbol{x}_i, i \in \{1, \ldots, S\}$.
Step 2: Define a "hypothesis" on the structure of the data in the embedding space χ, represented by "sampling" vectors $\boldsymbol{w}_k \in \chi$, $k \in \{1, \ldots, m\}, m \in \mathbb{N}$, and randomly initialize an "image" vector $\boldsymbol{\xi}_i \in \chi$, $i \in \{1, \ldots, S\}$ for each input vector $\boldsymbol{x}_i$. There is no principal limitation whatsoever of how such a sampling distribution could be chosen. Typical choices are a uniform distribution (e.g. in a 2D square) for structure-preserving visualization and the choice $\chi = \mathbb{R}^2$.
Step 3: Reconstruct the topology induced by the input data by moving the image vectors in the embedding space χ using the computational scheme of a topology-preserving mapping. The final positions of the image vectors $\boldsymbol{\xi}_i$ represent the output of the algorithm.

The image vectors $\boldsymbol{\xi}_i$ are incrementally updated by a sequential learning procedure. For this purpose, the neighborhood couplings between the input

data items are represented by a so-called cooperativity (or neighborhood) function, which is typical chosen as a Gaussian.

The overall complexity of the algorithm is quadratic in the number of data points. We refer to [44] for further details. Obviously, discriminative information can be included into XOM by substituting the distances computed in Step 1 by the discriminative distances as provided by LiRaM LVQ in Eq. 5.

2.7. Discriminative Maximum Variance Unfolding (MVU)

Maximum Variance Unfolding (MVU)[38] is a dimension reduction technique which aims at preservation if local distances. So distances between nearby high dimensional inputs $\{\boldsymbol{x}_i\}_{i=1}^S$ should match the distances between nearby low-dimensional outputs $\{\boldsymbol{\xi}_i\}_{i=1}^S$. Assume the inputs $\boldsymbol{x}_i$ are connected to their k nearest neighbors by rigid rods. The algorithm attempts to pull the inputs apart, maximizing the sum of their pairwise distances without breaking (or stretching) the rigid rods that connect the nearest neighbors. For the mathematical formulation let $\eta_{ij} \in \{0, 1\}$ denote whether inputs $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are k-nearest neighbors and solve the following optimization:

$$\begin{aligned}
&\text{Maximize } \sum_{ij} ||\boldsymbol{\xi}_i - \boldsymbol{\xi}_j||^2 \text{ subject to:} \\
&\quad (1)\ ||\boldsymbol{\xi}_i - \boldsymbol{\xi}_j||^2 = ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 \text{ for all } (i,j) \text{ with } \eta_{ij} = 1\ . \\
&\quad (2)\ \sum_i \boldsymbol{\xi}_i = 0\ .
\end{aligned}$$

This optimization is not convex, so the optimization is reformulated as a semi-definite program (SDP) by defining the inner product matrix $K_{ij} = \boldsymbol{\xi}_i \cdot \boldsymbol{\xi}_j$. The SDP over K can be written as:

$$\begin{aligned}
&\text{Maximize trace}(K) \text{ subject to:} \\
&\quad (1)\ K_{ii} - 2K_{ij} + K_{jj} = ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 \text{ for all } (i,j)\ . \\
&\quad (2)\ \sum_{ij} K_{ij} = 0\ . \\
&\quad (3)\ K \succeq 0 \text{ (positive semi-definiteness)}\ .
\end{aligned}$$

This SDP is convex and can be solved with polynomial-time guarantees. To include supervision in this dimension reduction technique the distance

defined by Eq. (5) can be used to determine the k nearest neighbors. Afterwards we simply apply the same optimization as original MVU. For our experiments we used the library for semi-definite programming called CSDP[1] and the MVU implementation provided by Kilian Q. Weinberger[2].

2.8. Further embedding techniques

We will compare the results obtained within this discriminative framework to a few standard embedding techniques. More precisely, we will display the results of linear discriminant analysis (LDA) [6] as a classical linear discriminative projection technology, t-Distributed SNE (t-SNE) as an extension of SNE which constitutes one of the most promising unsupervised projection techniques available today.

LDA constitutes a supervised projection and classification technique. Given data points and corresponding labeling, it determines a global linear map such that the distances within classes of projected points are minimized whereas the distances between classes of projected points are maximized. This objective can be formalized in such a way that an explicit analytical solution is obtained by means of eigenvalue techniques. It can be shown that the maximum dimension of the projection has to be limited to $C-1$, C being the number of classes, to give meaningful results. Hence, this method can only be applied for data sets with 3 or more classes. Further, the method is restricted to linear maps and it relies on the assumption that classes can be represented by unimodal clusters, which can lead to severe limitations in practical applications.

t-SNE constitutes an extension of SNE which achieved very promising visualization for a couple of benchmarks [37]. Unlike SNE, it uses a Student-t distribution in the projection space such that less emphasis is put on distant points. Further, it optimizes a slightly different cost criterion which leads to better results and an improved numerical robustness of the algorithm. The basic mapping principle of SNE and t-SNE, however, remains the same.

[1] http://infohost.nmt.edu/∼borchers/csdp.html
[2] http://www.weinbergerweb.net/Downloads/MVU.html

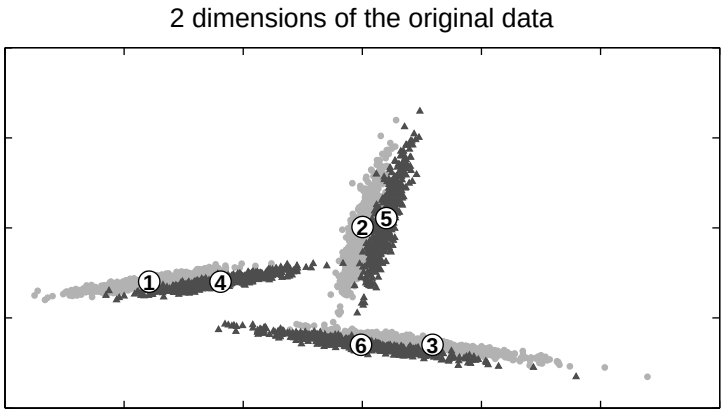

Figure 1: The two informational dimensions of the original Three Tip Star data set

3. Experiments

3.1. Three Tip Star

This artificial dataset consists of 3000 samples in $\mathbb{R}^{10}$ with two overlapping classes (C1 and C2), each forming three clusters as displayed in Fig. 1. The first two dimensions contain the information whereas the remaining eight dimensions contribute high variance noise. Following the advise "always try Principal Component Analysis (PCA) first"[3] we achieve a leave-one-out Nearest Neighbor (NN) error of 29% in the data set mapped to two dimensions (the result is shown in Fig. 2 left panel).

The NN error on the two-dimensional projections of all methods with either Euclidean or supervised adapted distance are shown in Fig. 3. In the figure, a "+" appended to the name of a method indicates the use of the learned distance, in addition the reduced target Dimension in matrix learning M is given. Localized LiRaM LVQ was trained for $t = 500$ epochs, with three prototypes per class and local matrices of target dimension $M = 2$. Each of the prototypes was initialized close to one of the cluster centers. Initial elements of Ω_j were generated randomly according to a uniform distribution in $[-1, 1]$ with subsequent normalization of the matrix. The learning rate for prototype vectors follows the schedule $\alpha_1(t) = 0.01/(1 + (t-1) \cdot 0.001)$. Metric learning starts at epoch $t = 50$ with learning rate $\alpha_2(t) = 0.001/(1 +$

[3]John A. Lee, private communication, 2009.

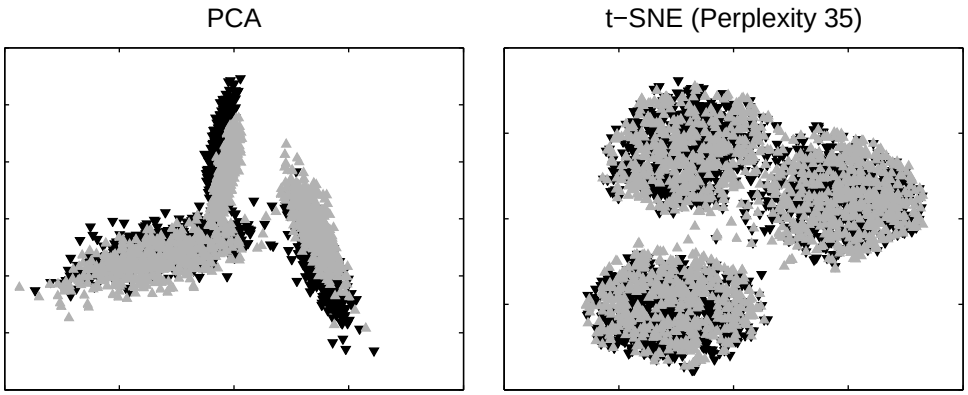

Figure 2: Example embeddings of the Three Tip Star data set for PCA and tSNE.

$(t-50)\cdot 0.0001)$.

XOM was trained for $t_{\max} = 50000$ iterations with a learning rate schedule $\epsilon(t) = \epsilon_1 \cdot \left(-\exp\left(\log\left(\frac{\epsilon_1}{\epsilon_2}\right)/t_{\max}\right)\cdot t\right)$ for the image vectors $\boldsymbol{\xi}$ with $\epsilon_1 = 0.9$ and $\epsilon_2 = 0.05$. The cooperativity function is chosen as Gaussian and like the learning rate $\epsilon(t)$ the variance σ is changed by an appropriate annealing scheme $\sigma(t) = \sigma_1 \cdot \left(-\exp\left(\log\left(\frac{\sigma_1}{\sigma_2}\right)/t_{\max}\right)\cdot t\right)$. The parameter σ_1 is set to round the maximum distance in the data space: 1500 and σ_2 is chosen as values between the interval $[10, 100]$. The sampling vectors are initialized randomly in 5 independent runs.

We repeat localized LiRaM LVQ with 10 independent random initializations. The resulting mean classification error in the three tip star data set is 9.7%. An example projection of the data according to Eq. (4) is displayed in Fig. 4 in the upper left panel. Note that the aim of the LiRaM LVQ algorithm is not to preserve topology or distances, but to find projections which separate the classes efficiently. Consequently, clusters four and six, for instance, may be merged in the projection, as they carry the same class label. Nevertheless, the relative orientation of all six clusters persists in the low-dimensional representation. The visualization shown in Fig. 4 (top right panel) displays the combination of the LiRaM LVQ projections with a charting step. Here, the parameter α for σ_j in Eq. (7) is set to 0.1. This value was found to give the best performance in a cross validation scheme for $k = 3$ nearest prototypes. Note that the quality of the projection is not affected by rotations or reflections, consequently the actual positions and orientations of

clusters can vary.

Fig. 3 displays the NN errors and standard deviations of local LiRaM LVQ as observed over 10 random initializations. From top to bottom the following methods are compared:

1. The NN errors in LiRaM LVQ projections based on Eq. (4). In particular, run 2 and 6 illustrate the problem that regions which are well separated in the original space can be projected onto overlapping areas in low dimension when local projection matrices Ω_j are employed naively. Frequently, however, a discriminative visualization is found, as an example the outcome of run 8 is shown in Fig. 4 (upper left panel).
2. The NN errors of the LiRaM LVQ projections followed by charting with different choices of the responsibilities, cf. σ_j Eq. (7). The x-axis corresponds to the factor α which determines σ_j from the mean distance of the k nearest prototypes. Graphs are shown for several values of k, and bars mark the standard deviations observed over the 10 runs. For large α and k the overlap of the local charts increases, yielding larger NN error in the final embedding. Small values of α, k lead to better projection results, an example is shown in Fig. 4 (upper right panel).
3. The NN errors of the XOM projections with different values of the parameter σ_2. The parameter $\sigma_1 = 1500$ is fixed to a value close to the maximum Euclidean distance observed on the data. The actual value of σ_2 appears to influence the result only mildly. The incorporation of the trained local distances improves the performance significantly. Example projections are shown in Fig. 4 (second row) using Euclidean distances (left panel) and for adaptive distance measure (right panel). The former, unsupervised version cannot handle this difficult data set satisfactorily, while supervised adaptation of the metric preserves the basic structure of the cluster data set.
4. The NN error of the Isomap projection with different numbers k of nearest neighbors taken into account. Also here the incorporation of the learned local distance reduces the NN error on the two-dimensional embedding significantly. The parameter k has to be large enough to ensure that a sufficient number of points is connected in the neighborhood graph. Otherwise several subgraphs emerge which are not connected and lead to many missing points in the final embedding. Appropriate example embeddings are shown in Fig. 4 in the third row, corresponding to Euclidean distance in the left panel and adaptive metrics in the

right panel. In the former, purely unsupervised case, the 3 main clusters are reproduced, but the classes are mixed. When the adaptive distance measure is used, the cluster structure is essentially lost, but the two classes remain separated.

5. The NN errors of the LLE embedding for various numbers k of nearest neighbors considered. LLE displays very limited performance in this data set, hardly any structure is preserved. Even the incorporation of the learned distance measure does not lead to significant improvement, in general. Only for very small values of k the NN error decreases in comparison with the usage of the Euclidean distance. LLE tends to collapse large portions of data onto a single point when the target dimension is too low. Hence, even a small NN error may not indicate a good and interpretable visualization. The best embeddings are shown in Fig. 4 in the forth row.
6. The NN errors of SNE and t-SNE are slightly better than the other unsupervised methods. Both methods preserve the main cluster structure, but not the class memberships.
   Like already observed with Isomap$^+$ also with the supervised version SNE (SNE$^+$) the cluster structure is essentially lost, but the two classes are separated as much as possible and a remarkable increase in the NN error of the embedded data is observed. Example embeddings are shown in Fig. 4 (fifth row) and for t-SNE in Fig. 2 right panel.
7. The NN errors of MVU are comparable to the SNE and t-SNE results. Like them the main cluster structure is visible, but not the class memberships. In the supervised variant MVU$^+$ the cluster structure is essentially lost as observed with Isomap$^+$ and SNE$^+$ too, but the two classes are separated relatively well. This leads to a remarkable decrease in the NN error of the embedded data points. The best embeddings are shown in Fig. 4 at the bottom row.

Note that, due to the presence of only two classes, standard linear discriminance analysis (LDA) would yield a projection to one dimension only. We have also applied kernel PCA with Gaussian kernel and different values of σ. We have obtained only poor NN errors on the embedded data with a best value of about 41%.

As expected, purely unsupervised methods preserve hardly any class structure in the obtained projections. For several methods, however, the performance with respect to discriminative low-dimensional representation

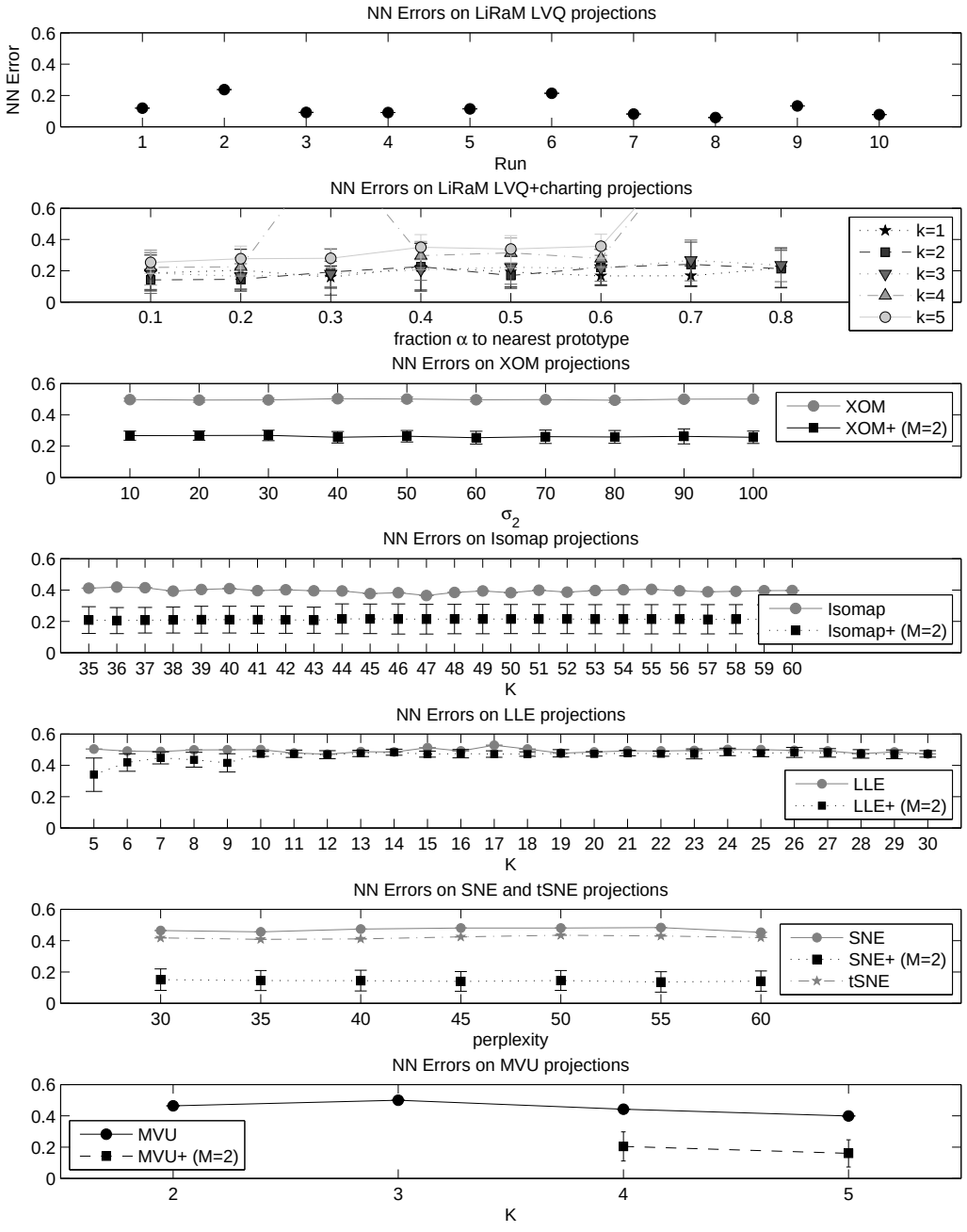

Figure 3: NN Errors in the Three Tip Star data set for different methods and parameters. A "+" appended to the name of the method indicates incorporation of local LiRaM LVQ distances with rank M matrices.

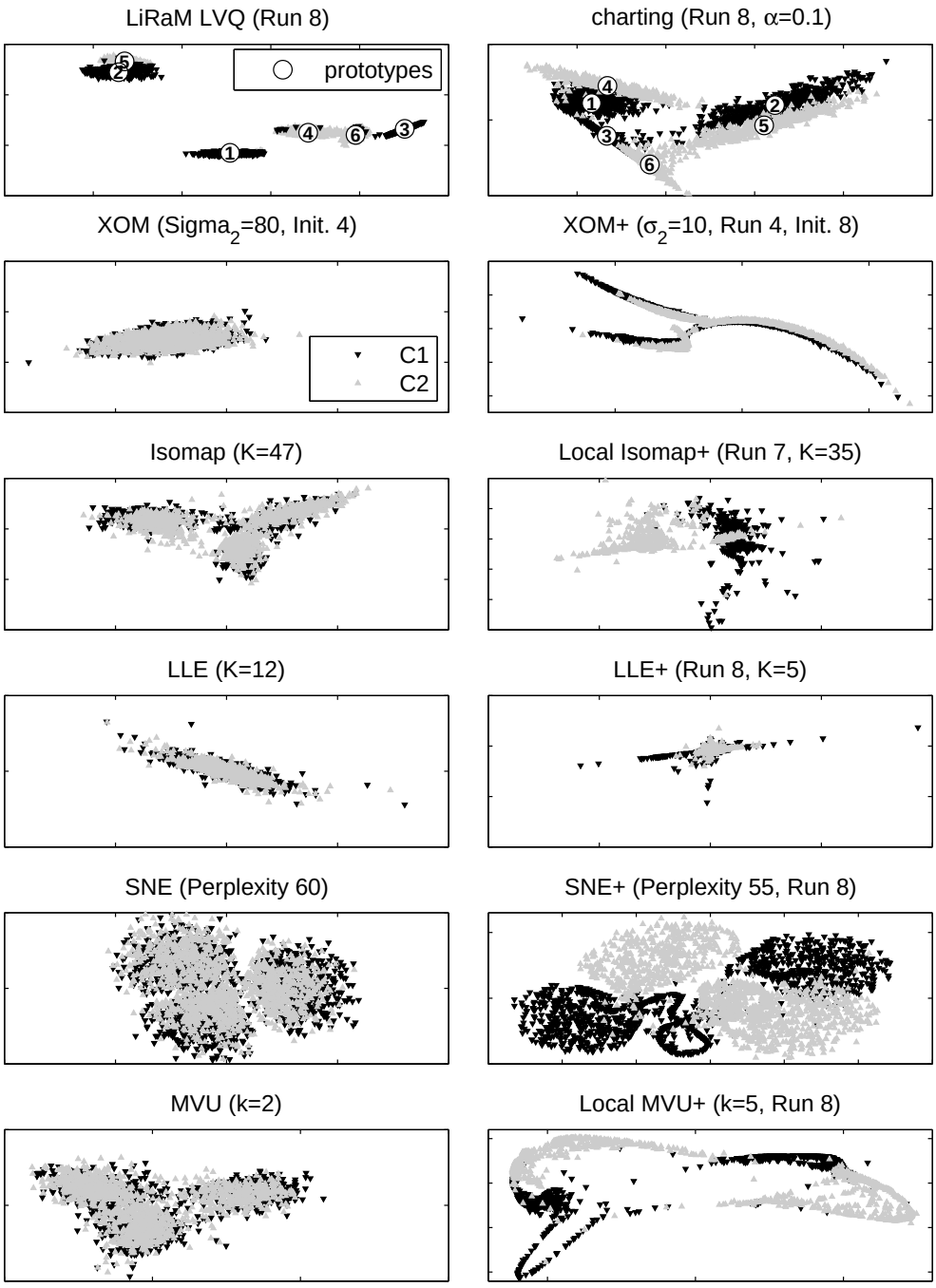

Figure 4: Example embeddings of the Three Tip Star data set for different methods. A "+" appended to the name of the method indicates incorporation of local LiRaM LVQ distances with rank M matrices.

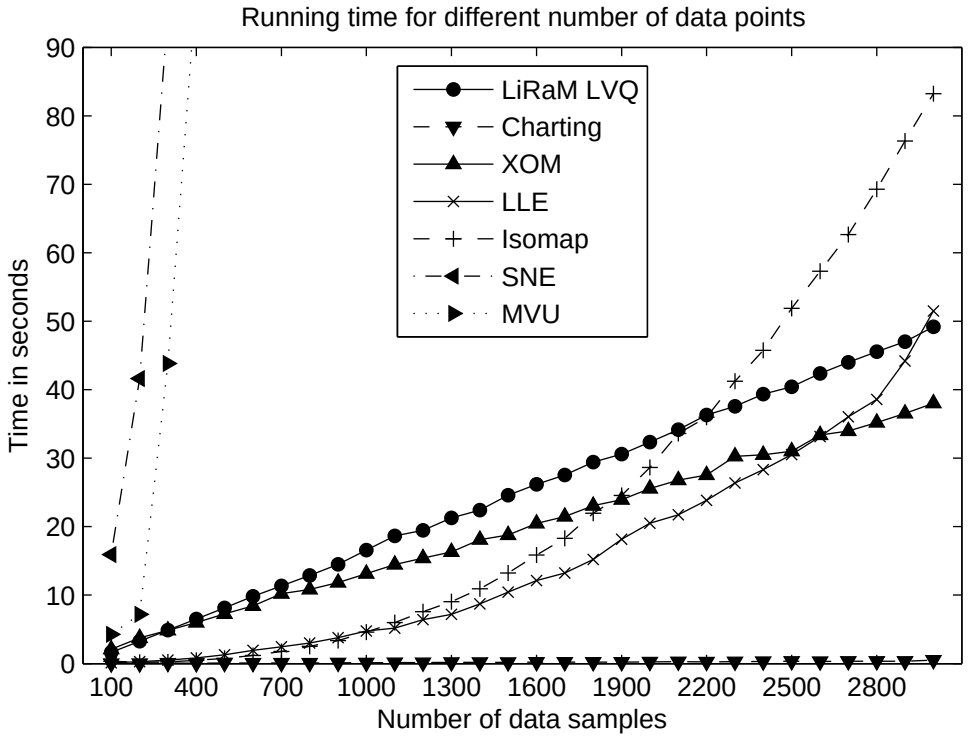

Figure 5: The running time of different dimension reduction methods depending on the number of samples to embed.

can be improved dramatically by taking into account label information in the local distance measures.

Figure 5 shows the computation times vs. the number of points to be embedded of different dimension reduction techniques on the Three Tip Star data set. We only measure the time necessary to embed the data after learning the local metrics with LiRaM LVQ. The algorithms were performed on the same Windows XP 32bit version machine[4] using Matlab R2008b. The LiRaM LVQ algorithm was applied using six prototypes and 100 epochs. The other parameters were chosen as mentioned above. The charting technique uses the six local linear projections provided by the LVQ approach with responsibilities computed by Eq. (6). XOM is trained for 1500 steps and above mentioned parameters, LLE uses $k = 35$, Isomap $k = 35$ and MVU

[4]Intel(R) Core(TM)2 Quad CPU Q6600 @2.40GHz, 2.98 GB of RAM

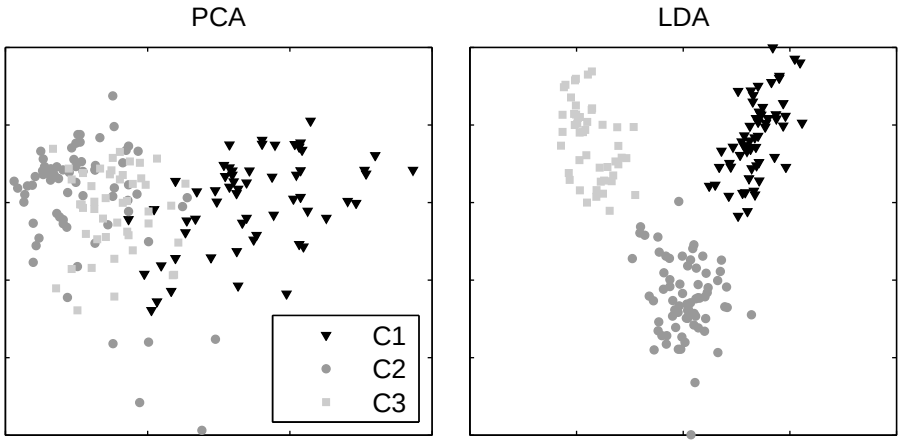

Figure 6: Example embeddings of the Wine data set for PCA and LDA.

$k = 3$ nearest neighbors. SNE was performed with a perplexity of 30. The LVQ based approach, charting and XOM show a linear relationship between the number of points and the necessary computation time, whereas the other methods show quadratic or even worse complexity.

3.2. Wine data set

The wine data from [46] available at [47] contains 178 samples in 13 dimensions divided in three classes. As proposed in [48] we first transformed the data to have zero mean and unit variance. Maximum Likelihood Estimation (MLE) [49] approximate the intrinsic dimension to 4. We set the reduced target dimension to two. Unsupervised PCA achieves a leave-one-out Nearest Neighbor (NN) error of 28% in the mapped data set. In comparison, supervised Linear Discriminant Analysis (LDA) [6] leads to a relatively small NN error of 1%. Fig. 6 shows the two-dimensional representations of the data set obtained by PCA and LDA, respectively.

Localized LiRaM LVQ was trained for $t = 300$ epochs, with one prototype per class. Each prototype was initialized close to class centers, elements of the matrices Ω_j were drawn with uniform density from the interval $[-1, 1]$ with subsequent normalization. The learning rate for prototype updates follows the schedule $\alpha_1(t) = 0.1/(1 + (t - 1) \cdot 0.01)$; metric learning starts at epoch $t = 30$ with the learning rate $\alpha_2(t) = 0.01/(1 + (t - 50) \cdot 0.001)$. We run the localized LiRaM LVQ 10 times with random initializations and with rank

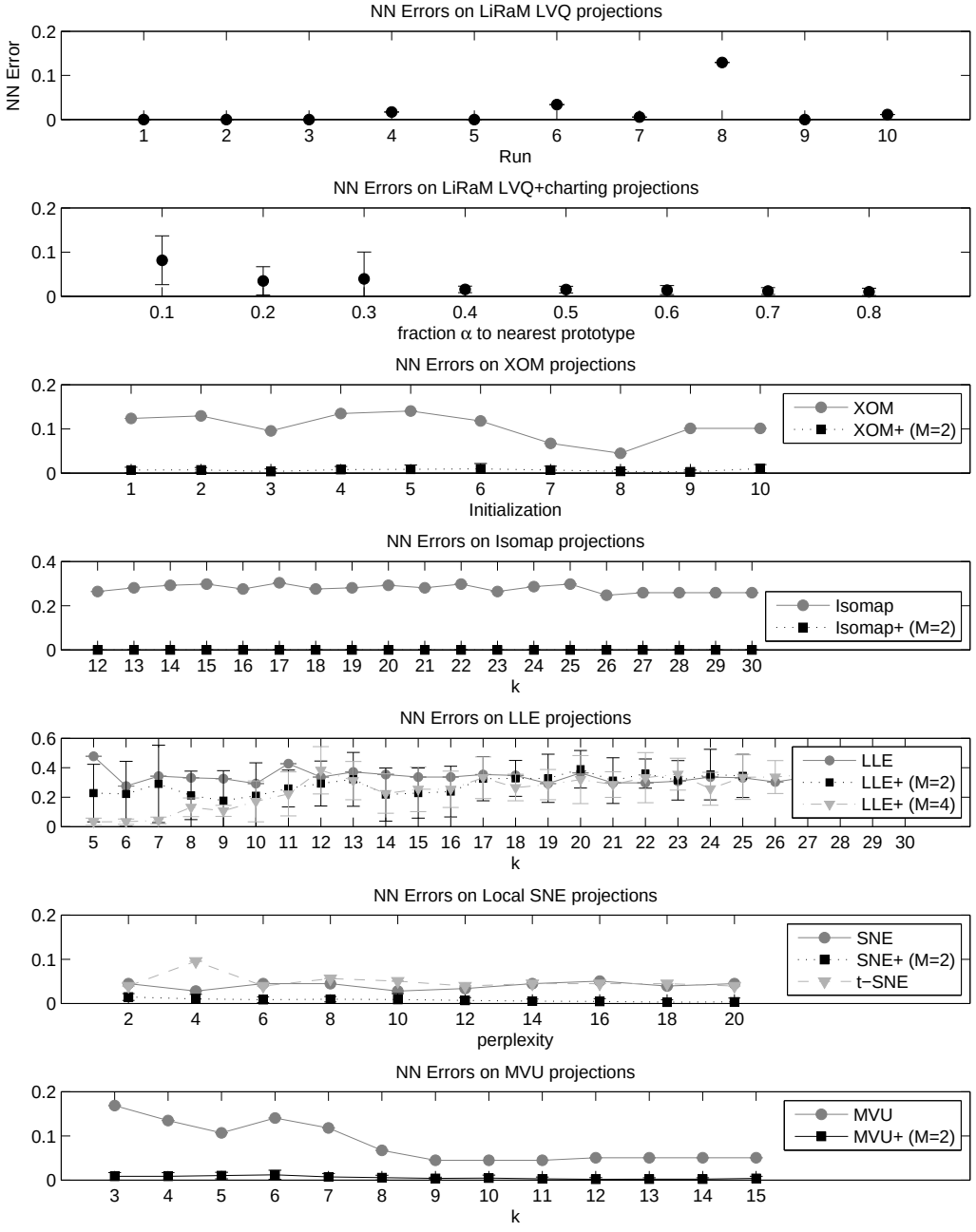

Figure 7: NN Errors in the Wine data set for different methods and parameters. A "+" appended to the name of the method indicates incorporation of local LiRaM LVQ distances with rank M matrices.

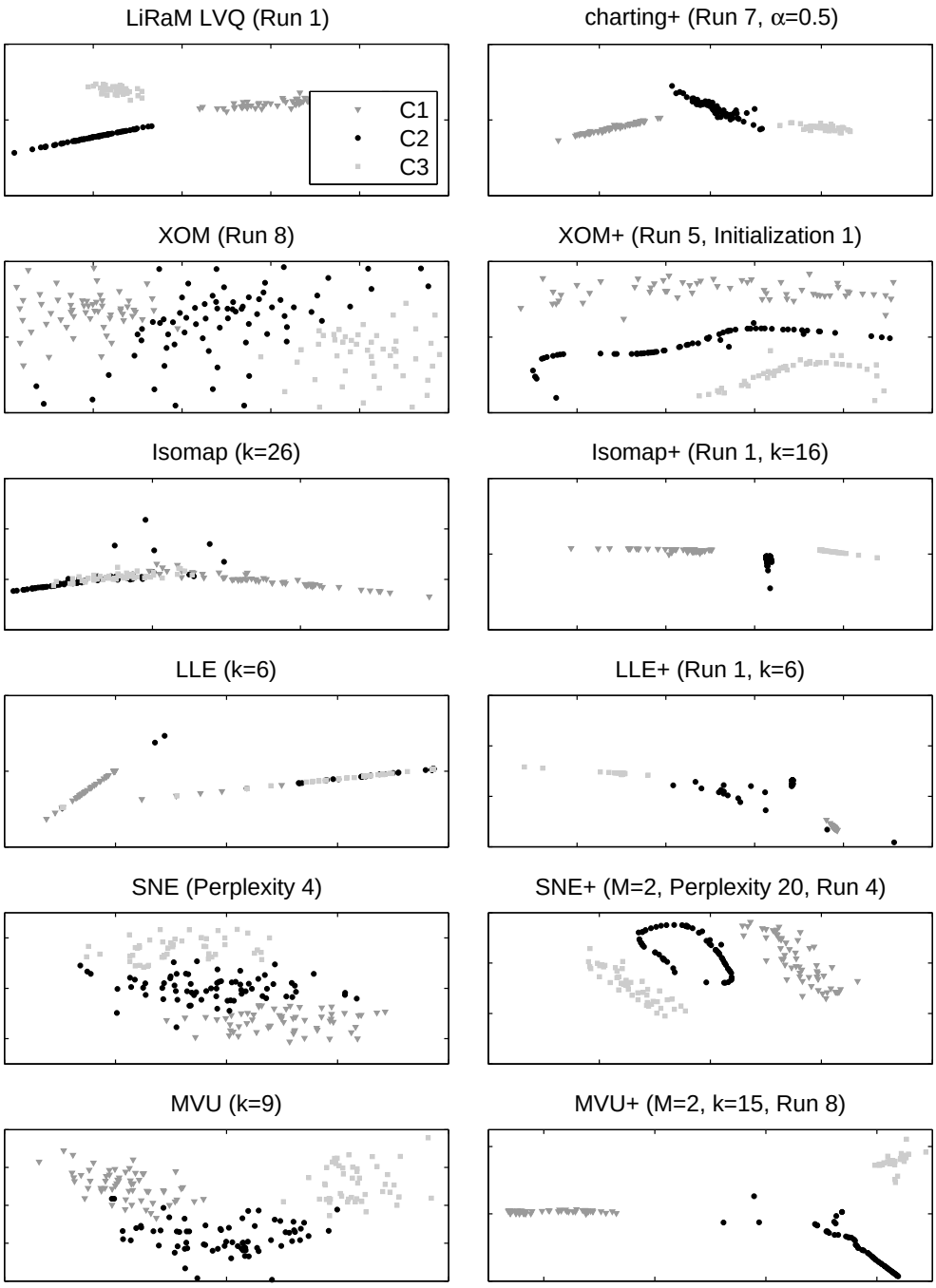

Figure 8: Example embeddings of the Wine data set for different methods. A "+"-sign appended to the name of the method indicates incorporation of local LiRaM LVQ distances with rank M matrices.

$M = 2$ and $M = 4$ of the relevance matrices, respectively. In all runs we observe 100% correct classification for this data set.

The resulting matrices are used to embed the data into the two-dimensional space. In order to compare the different approaches we compute the NN errors in the projected data under various parameter settings, results are shown in Fig. 7. The incorporation of trained distances in some unsupervised methods are indicated by a "+" appended to the name, together with the maximum rank M. In the direct LVQ-based mapping, Eq. (4), two prototypes project into the same area in some of the runs, but most runs result in a clear separation of the three classes. The charting technique is combined with the three local projections obtained from the localized LiRaM LVQ ($M = 2$) and computed with various parameters α used to fix the responsibilities (see Eq. (6)). A reasonable overlap of the local projections is required: If α is chosen too small the NN error displays large variations in the runs and the classes overlap. For this data set a value of $\alpha = 0.4$ is sufficiently large to yield discriminative visualizations.

XOM was trained like with the previous data set for $t_{\max} = 50000$ iterations with the same learning rate schedule for ϵ and σ. The parameter σ_1 is set to 2 and σ_2 to 0.15. The sampling vectors are initialized randomly in 10 independent runs. The results of XOM and XOM in combination with adaptive local distances are analogous to those for the Three Tip Star data. The improvement due to the incorporation of label information through the distance measure is even more significant, the method yields very small NN errors in the Wine data set.

The k-Isomap with Euclidean distance performs worse on this data set with an NN error of about 30%. With the incorporation of the learned distance measure and a sufficiently large neighborhood value k all mappings separate the classes very well. For smaller values of k the neighborhood graph is not connected. In the worst case the procedure yields three unconnected subgraphs, where only samples are connected which belong to the same prototype. When all samples are connected the approach is very robust and shows no variation with respect to the LVQ run.

The performance of LLE depends strongly on the number k of nearest neighbors taken into account. For large k the advantage of using a supervised learned distance measure essentially vanishes. The variations between different runs are particularly pronounce for rank $M = 2$ and no significance improvement over the purely unsupervised LLE is achieved. However, for small k (e.g. $k = 5, 6, 7$) and with rank $M = 4$ very low NN errors are

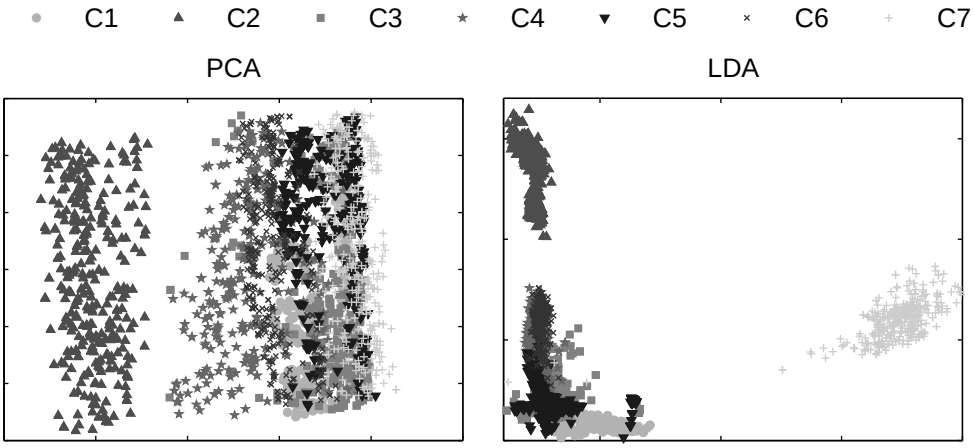

Figure 9: Example embeddings of the Segmentation data set for PCA and LDA.

obtained.

The SNE and t-SNE show already in the unsupervised versions good results as shown in Fig. 7. The NN error is not that much dependent on the chosen perplexity, only slight changes can be observed. With the incorporation of the learned distance measure the visualizations can be improved further and the dependence on the perplexity is even less.

The unsupervised MVU showed a strong dependence on the parameter k, the number of neighbors taken into account. With a sufficient big k the algorithm show already good results when it is used in an unsupervised way. The incorporation of the class labels however shows only a weak dependence on the parameter k and in most of the results the classes are perfectly separated. For visual comparison we pick the best mappings of each method and display them in Fig. 8.

3.3. Segmentation

The segmentation data set (available at the UCI repository [47]) consists of 19 features which have been constructed from randomly drawn regions of 3×3 pixels in a set of 7 manually segmented outdoor images. Every sample is assigned to one of seven classes: brickface, sky, foliage, cement, window, path and grass (referred to as C1, ..., C7). The set consists of 210 training points with 30 instances per class and the test set comprises 300 instances per class, resulting in 2310 samples in total. We did not use the features (3,4,5) as they display zero variance over the data set. We do not preprocess or normalize

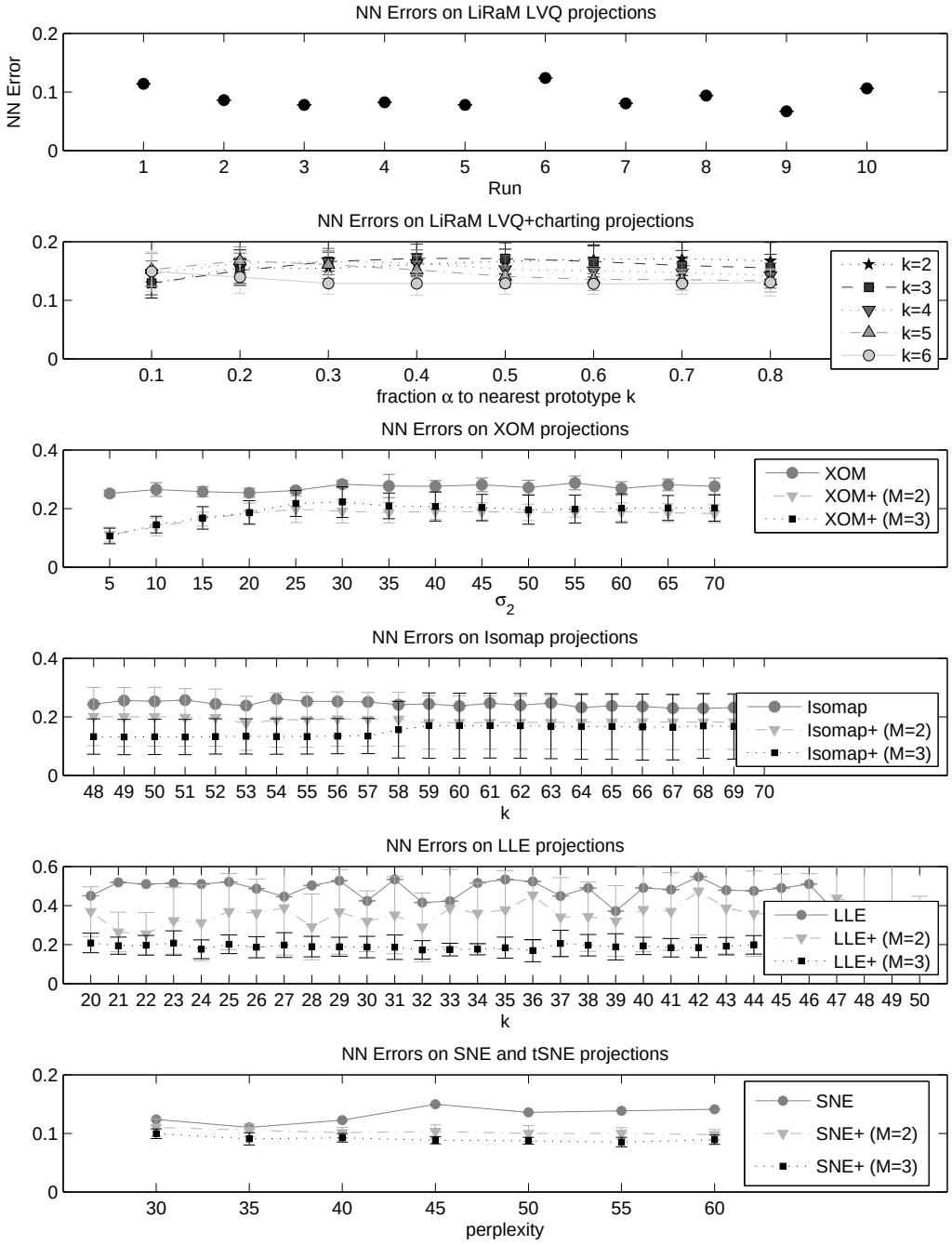

Figure 10: NN Errors on the Segmentation data set for different methods and parameters. A '+" appended to the name of the method indicates incorporation of local LiRaM LVQ distances with rank M matrices.

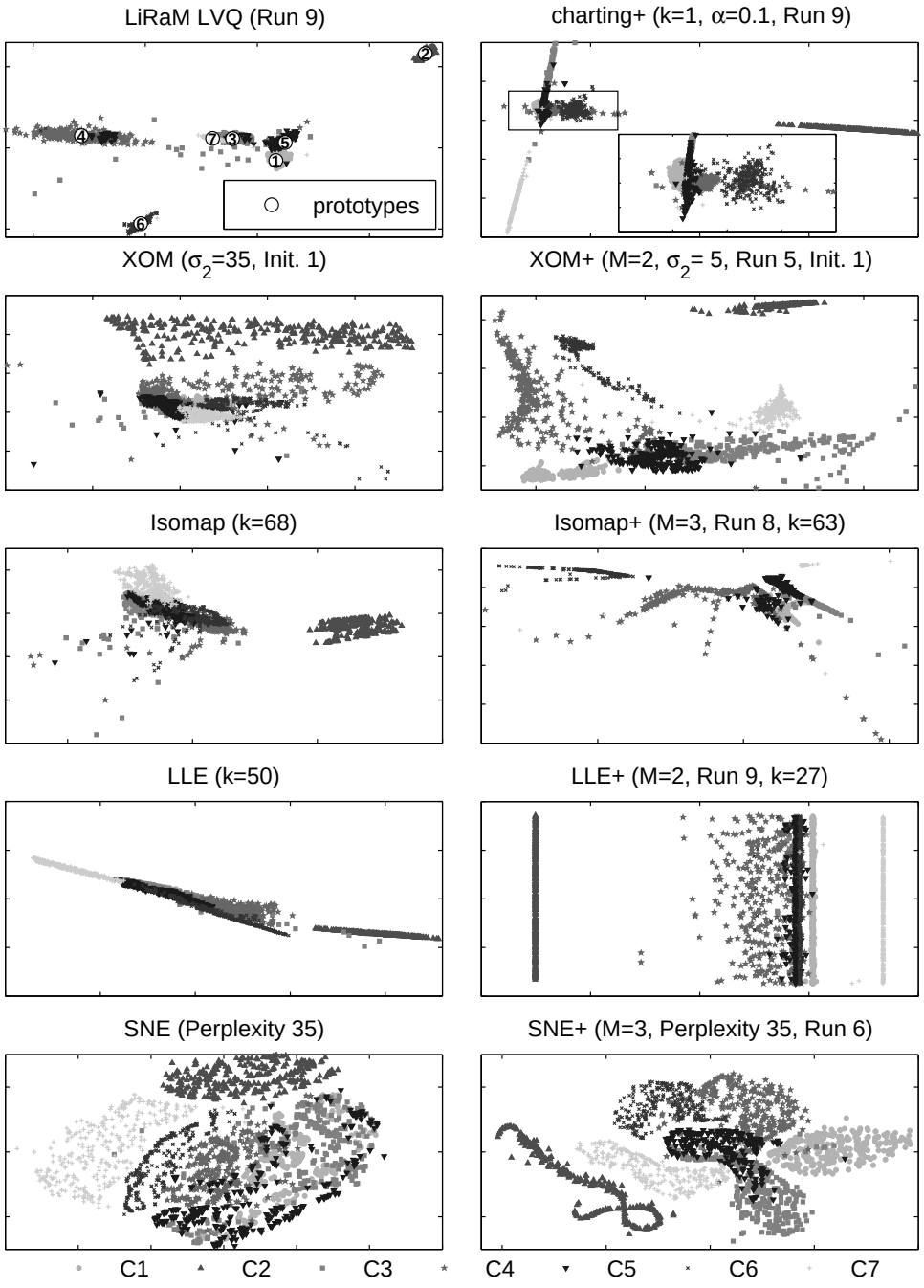

Figure 11: Example embeddings of the Segmentation data set for different methods. A "+" appended to the name of the method indicates incorporation of local LiRaM LVQ distances with rank M matrices. The inset in the upper right panel magnifies the region to which four of the classes are mapped.

the data because Isomap, LLE, and SNE displayed better performance on original data. For t-SNE different magnitudes of features may constitute a problem, which we observed by a very high NN error in comparison with the other methods, and we would expect it to perform better on, e.g., z-transformed data. An ML estimation yields an intrinsic dimension of about 3, so we use rank limits of $M \in \{2, 3\}$ for the computation of the local distances in this data set.

Localized LiRaM LVQ was trained for $t = 500$ epochs, with one prototype per class. Each prototype was initialized close to class center, and elements of the matrices Ω_j are drawn randomly from $[-1, 1]$ according to a uniform density with subsequent normalization of the matrices. The learning rate for prototypes follows the schedule $\alpha_1(t) = 0.01/(1 + (t - 1) \cdot 0.001)$. Metric adaptation starts at epoch $t = 50$ with learning rates $\alpha_2(t) = 0.001/(1 + (t - 50) \cdot 0.0001)$. We run localized LiRaM LVQ 10 times with random initialization and with a rank limit of $M = 2$ and $M = 3$, respectively. For $M = 2$ we achieve a mean classification error of about 8% in all runs and with $M = 3$ the mean classification error is 7%.

LDA yields a classification error of approx. 20% for a projection into two dimensions while unsupervised PCA displays a NN error of 31%. The obtained NN errors are shown in Fig. 10 and some example visualizations are given in Fig. 11.

The quality of direct LiRaM LVQ projections vary from run to run. One favorable projection is shown in Fig. 11 in the first row on the left side. The classes C2 and C6 are well separated with large distances from the other classes. Also, most samples of C4 and C1 are clustered properly, while class C3 is spread and overlaps with class C7. This outcome is not too surprising, since C3 and C7 correspond to foliage and grass, respectively, two classes that may be expected to have similar characteristics in feature space.

In the combination with a charting step results are rather robust with respect to the parameter settings (α, k). Here, the best result is achieved with $\alpha = 0.1$ and $k = 1$ (Fig. 11, top right panel). Again, three classes are well separated from the others. The remaining four classes are projected into a relatively small area. As can be seen in the inset, three of these classes are very close: window, brickface, and cement. Again, similar properties in feature space can be expected for these classes.

XOM was trained for $t_{\max} = 50000$ iterations with the same learning rate schedule for ϵ and σ like for the other data sets. We set the parameters to $\epsilon_1 = 0.9$, $\epsilon_2 = 0.05$ and σ_1 to nearly the maximum distance in the data space:

1500 and σ_2 is chosen as values between the interval $[5, 70]$. The sampling vectors are initialized randomly in 5 independent runs. In the application of XOM we observe once more a clear improvement when incorporating the adaptive local metrics obtained in LiRaM LVQ. Example projections are shown in Fig. 11 (second row).

For Isomap a minimum value of $k \geq 48$ is necessary to obtain fully connected neighborhood graphs and, hence, embed all points. The incorporation of adaptive local distances leads to a clear improvement of the NN error in the mapping.

As expected, a low rank M of the local matrices results in inferior NN errors if M is smaller than the intrinsic dimension of the data. When incorporating adaptive distances with very large k, a fully connected graph can be obtained and all data are mapped. However, then, closer classes would highly overlap in the projections and the visualization would not be discriminative. If, on the other hand, a smaller k is chosen, some of the classes are absent in the graph and, consequently, in the visualization. As a consequence of this effect, in Fig. 11 (third row, right panel) class C2 subgraph is absent.

Like in the previous examples, LLE performs relatively poor. The NN error can be decreased by using adaptive distances but points tend to be collapsed in the projection due to the discriminative nature of the distance measure. Most visualizations with relatively low NN errors display an almost linear arrangement of all classes, cf. Fig. 11 (fourth row, left panel). An example visualization after incorporation of adaptive metrics is shown in the right panel. While the visualization appears to be be better, qualitatively, the above mentioned basic problem of LLE persists.

The last row of Fig. 11 displays the two-dimensional representations provided by SNE and SNE$^+$ for perplexities in the interval [30 60]. The unsupervised variant performs already quite well, but the incorporation of the learned local distances improves it even further especially for higher perplexities and bigger values for the limited rank M of the LiRaM LVQ algorithm (see Fig. 10).

Classes C2 (sky) and C7 (grass) are obviously separable by all applied methods, both unsupervised and supervised. On the other hand, the discrimination of classes C4 (foliage) and C5 (window) appears to be difficult, in particular in unsupervised dimension reduction.

We could not evaluate MVU on this data set, because this would require the costly incorporation of in minimum $k = 46$ neighbors. It appears, that a part of the data is already well separated, so that the neighborhood graph is

not connected with smaller values of k. The provided code demands a fully connected graph, so the number of constraints of the SDP becomes too large to be solved in reasonable time and needs more memory than we have.

3.4. USPS Digits

The USPS[5] dataset consists of images of hand written digits of a resolution of 16×16 pixel. We normalized the data to have zero mean and unit variance and used a test set containing 200 observations per class. Since it is a digit recognition task , we have the classes $\in [0, \ldots, 9]$ resulting in 2000 samples for the embedding. The NN errors of all compared methods are shown in Fig. 12.

Localized LiRaM LVQ was trained for $t = 500$ epochs, with one prototype per class and the same initialization scheme for the prototypes and matrices, learning rates and learning schedules like explained in section 3.3. The direct LiRaM LVQ projections separate the classes nearly perfectly and one favorable projection is shown in Fig. 13 in the first row on the left side. In the combination with a charting step the best result is achieved with $\alpha = 0.1$ and $k = 2$ (Fig. 13, top right panel). Four classes appear to be squeezed together, but the overlap is still small if zoomed.

XOM was trained in the same way like mentioned in section 3.3 with σ_2 chosen as values between the interval [0.01,2]. The incorporation of the adaptive local metrics obtained in LiRaM LVQ once more improve the results of the XOM dramatically. Example projections are shown in Fig. 13 (second row).

For Isomap the incorporation of adaptive local distances improves the NN error in the mapping. Like mentioned with the other data sets some data points appear to be too separated from the others if the local distances are used, so the mapping may miss them with a small neighborhood parameter k. Like in the previous examples, LLE performs relatively poor, but can be enhanced by using the local dissimilarities given by LiRaM LVQ (fig. 13, fourth row).

SNE performs relatively well, but t-SNE showed a remarkable better NN error on this data set. Still the class structure is hardly recognizable on the unsupervised mapping, while it becomes clear if the local distances are incorporated (Fig. 13, fifth row, right panel). The supervised SNE$^+$ results

[5]United States Postal Service (U.S. Postal Service)

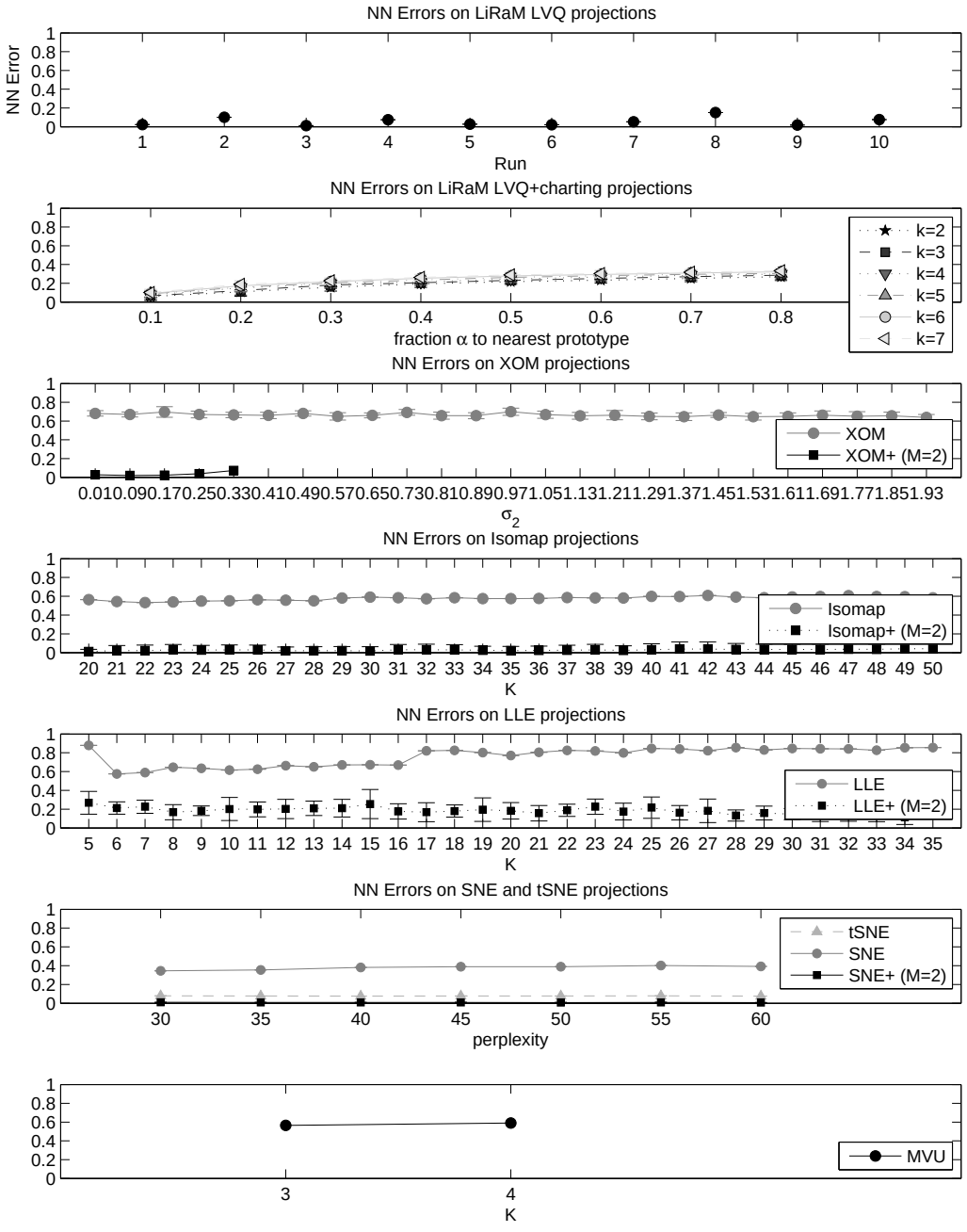

Figure 12: NN Errors on the USPS data set for different methods and parameters. A '+" appended to the name of the method indicates incorporation of local LiRaM LVQ distances with rank M matrices.

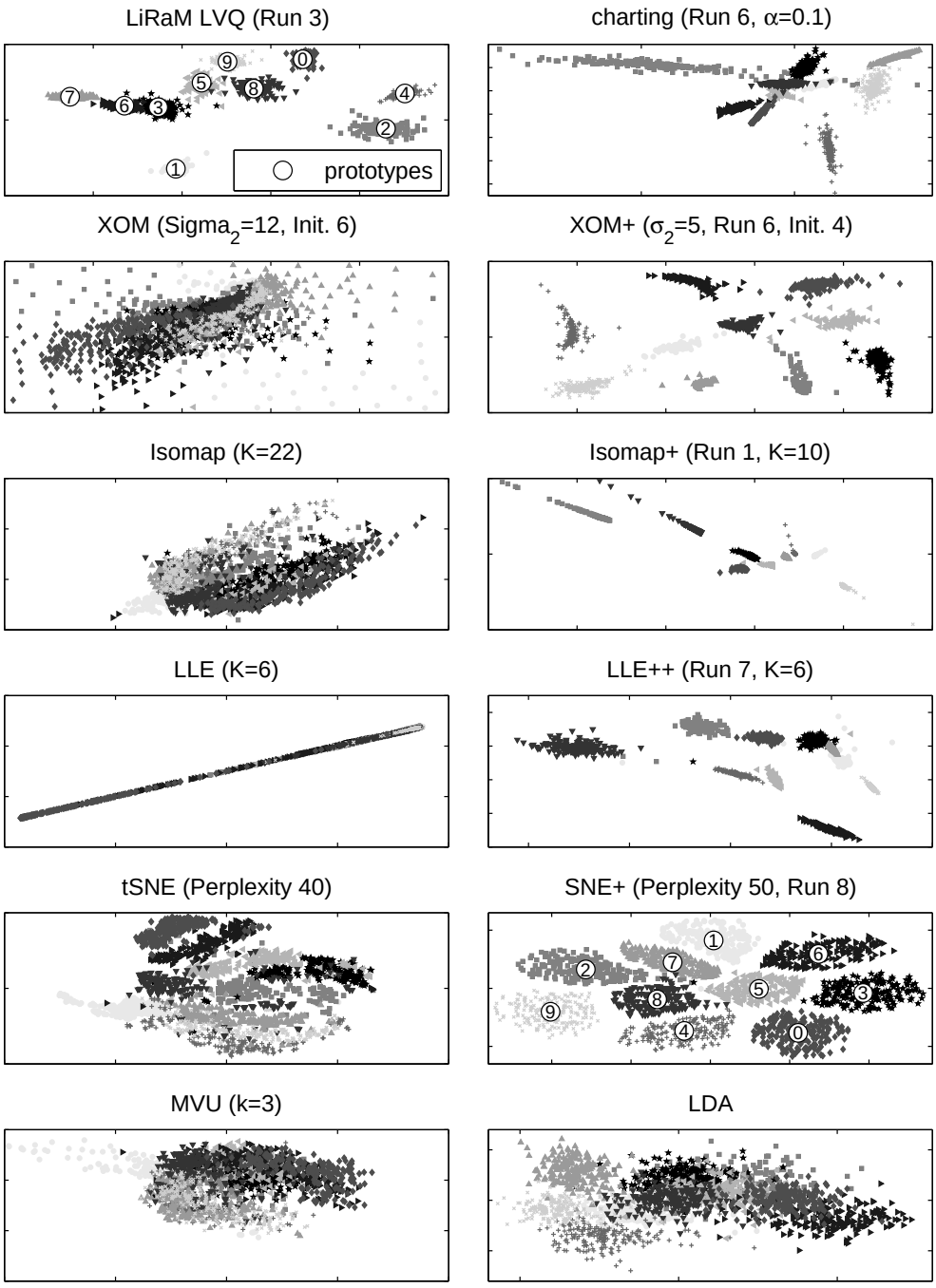

Figure 13: Example embeddings of the USPS data set for different methods. A "+" appended to the name of the method indicates incorporation of local LiRaM LVQ distances with rank M matrices.

Table 1: NN errors (and Standard deviation) on the different data sets.

Method	3 Tip Star	wine	segmentation	USPS
LiRaM LVQ	0.06 (0.0)	0.00 (0.0)	0.07 (0.0)	0.01 (0.0)
charting	0.14 (0.1)	0.01 (0.0)	0.13 (0.0)	0.06 (0.0)
XOM	0.49 (0.0)	0.04 (0.0)	0.25 (0.0)	0.64 (0.0)
XOM+(M=2)	0.25 (0.0)	0.00 (0.0)	0.11 (0.0)	0.02 (0.0)
XOM+(M=3)	-	-	0.11 (0.0)	-
Isomap	0.36 (0.0)	0.25 (0.0)	0.23 (0.0)	0.53 (0.0)
Isomap+(M=2)	0.20 (0.1)	0.00 (0.0)	0.18 (0.1)	0.01 (0.0)
Isomap+(M=3)	-	-	0.13 (0.1)	-
LLE	0.47 (0.0)	0.28 (0.0)	0.36 (0.0)	0.57 (0.0)
LLE+(M=2)	0.34 (0.1)	0.18 (0.2)	0.25 (0.1)	0.11 (0.1)
LLE+(M=3)	-	0.03 (0.0)	0.19 (0.0)	-
SNE	0.45 (0.0)	0.03 (0.0)	0.11 (0.0)	0.34 (0.0)
SNE+(M=2)	0.14 (0.1)	0.00 (0.0)	0.10 (0.0)	0.01 (0.0)
SNE+(M=3)	-	-	0.09 (0.0)	-
t-SNE	0.41 (0.0)	0.04 (0.0)	0.85 (0.0)	0.08 (0.0)
MVU	0.40 (0.0)	0.04 (0.0)	-	0.56 (0.0)
MVU+(M=2)	0.16 (0.1)	0.00 (0.0)	-	-

in 10 nicely recognizable clusters.

The last row of Fig. 13 displays the two-dimensional representations provided by MVU and LDA. MVU does not perform very well on this data set and LDA yields a classification error of 35% for a projection into two dimensions. We could not apply MVU with incorporation of the local distances provided by LiRaM LVQ, because the classes are separated so well in this case that a huge value of nearest Neighbors k would be necessary to get a connected graph.

4. Conclusions

We have introduced the concept of discriminative nonlinear data visualization based on local matrix learning. Unlike unsupervised visualization schemes, the resulting techniques focus on the directions which are locally of

particular relevance for an underlying classification task such that the information which is important for this additional label information is preserved by the visualization as much as possible. Interestingly, local matrix learning gives rise to auxiliary information which can be integrated into visualization techniques in different form: as local discriminative coordinates of the data points for charting techniques and similar methods, as global metric information for XOM, SNE, MDS, and the like, or as local neighborhood information for LLE, Isomap, MVU and similar schemes. We have introduced these different paradigms and we exemplary presented the behavior of these schemes for six concrete visualization techniques, namely charting, LLE, Isomap, XOM, SNE and MVU. An extension to further methods such as t-SNE, diffusion maps, etc. could be done along the same lines.

Interestingly, the resulting methods have quite different complexity: while charting uses the fact that information is compressed in the prototypes resulting in an only linear scheme depending on the number of data, LLE, SNE, and Isomap end up with quadratic or even cubic complexity. Further, charting techniques and similar provide the only methods in this collection which yield an explicit embedding map rather than an embedding of the given points only. The behavior of the resulting discriminative visualization techniques has been investigated in one artificial and three real life data sets. The best results for all methods and data sets are summarized in Table 1. According to the different objectives optimized by the visualization techniques, the results are quite diverse and no single method which is optimum for every case can be identified. In general, discriminative visualization as introduced in this paper improves all the corresponding unsupervised methods and also alternative state-of-the-art schemes such as t-SNE. Further, the techniques presented in this paper are superior to discriminative LDA which is restricted to linear embedding. It seems that charting offers a good choice in many cases, in particular since it is a method with only linear effort which provides an explicit embedding map.

Interestingly, a direct projection of the data by means of the local linear maps of LiRaM LVQ displays good results in many cases, although an appropriate coordination of these maps cannot be guaranteed in this technique. It seems promising to investigate the possibility to introduce the objective of valid coordination of the local projections directly into the LiRaM LVQ learning scheme. This issue as well an exhaustive comparison of more extensions of unsupervised methods (such as t-SNE) to incorporate discriminative information are the subject of ongoing work.

This work was supported by the "Nederlandse organisatie voor Wetenschappelijk Onderzoek (NWO)" under project code 612.066.620.

References

[1] W. J. Frawley, G. Piatetsky-Shapiro, C. J. Matheus, Knowledge Discovery in Databases: an Overview, AAAI / MIT Press, Cambridge, 1–27, eds. G. Piatetsky-Shapiro and W. Frawley, 1991.

[2] J. Lee, M. Verleysen, Nonlinear dimensionality reduction, Springer, 3rd edn., 2007.

[3] L. J. P. van der Maaten, E. O. Postma, H. J. van den Herik, Dimensionality Reduction: A Comparative Review, URL http://ticc.uvt.nl/~lvdrmaaten/Laurens_van_der_Maaten/Matlab_Toolbox_for_Dimensionality_Reduction_files/Paper.pdf, published online., 2007.

[4] D. A. Keim, Information Visualization and Visual Data Mining, IEEE Transactions on Visualization and Computer Graphics 8 (1) (2002) 1–8, ISSN 1077-2626, doi: http://doi.ieeecomputersociety.org/10.1109/2945.981847.

[5] K. Bunte, P. Schneider, B. Hammer, F.-M. Schleif, T. Villmann, M. Biehl, Discriminative Visualization by Limited Rank Matrix Learning, Tech. Rep. MLR-03-2008, Leipzig University, URL http://www.uni-leipzig.de/~compint/mlr/mlr_03_2008.pdf, 2008.

[6] K. Fukunaga, Introduction to Statistical Pattern Recognition, 2nd edn. (Computer Science and Scientific Computing Series), Academic Press, ISBN 0122698517, 1990.

[7] J. Faith, R. Mintram, M. Angelova, Targeted Projection Pursuit for Visualising Gene Expression Data Classifications, Bioinformatics 22 (2006) 2667–2673.

[8] J. Peltonen, J. Goldberger, S. Kaski, Fast Discriminative Component Analysis for Comparing Examples, NIPS .

[9] T. Villmann, B. Hammer, F.-M. Schleif, T. Geweniger, W. Hermann, Fuzzy classification by fuzzy labeled neural gas, Neural Networks 19 (6-7) (2006) 772–779.

[10] J. Peltonen, A. Klami, S. Kaski, Improved Learning of Riemannian Metrics for Exploratory Analysis, Neural Networks 17 (2004) 1087–1100.

[11] P. Kontkanen, J. Lahtinen, P. Myllymäki, T. Silander, H. Tirri, Supervised model-based visualization of high-dimensional data, Intell. Data Anal. 4 (3,4) (2000) 213–227, ISSN 1088-467X.

[12] T. Iwata, K. Saito, N. Ueda, S. Stromsten, T. L. Griffiths, J. B. Tenenbaum, Parametric Embedding for Class Visualization., Neural Computation 19 (9) (2007) 2536–2556, URL http://dblp.uni-trier.de/db/journals/neco/neco19.html#IwataSUSGT07.

[13] G. Baudat, F. Anouar, Generalized Discriminant Analysis Using a Kernel Approach, Neural Computation 12 (10) (2000) 2385–2404.

[14] B. Kulis, A. Surendran, J. Platt, Fast low-rank semidefinite programming for embedding and clustering, in: Eleventh International Conference on Artifical Intelligence and Statistics, AISTATS 2007, 2007.

[15] N. Vasiloglou, A. Gray, D. Anderson, Scalable semidefinite manifold learning, in: IEEE Workshop on Machine Learning for Signal Processing, 2008. MLSP 2008, 368–373, 2008.

[16] R. Collobert, F. Sinz, J. Weston, L. Bottou, Trading convexity for scalability, in: Proceedings of the 23rd international conference on Machine learning, ACM New York, NY, USA, 201–208, 2006.

[17] L. Song, A. J. Smola, K. Borgwardt, A. Gretton, Colored Maximum Variance Unfolding, in: J. C. Platt, D. Koller, Y. Singer, S. Roweis (Eds.), 21th Neural Information Processing Systems Conference, MIT Press, Cambridge, MA, USA, 1385–1392, URL `http://books.nips.cc/papers/files/nips20/NIPS2007_0492.pdf`, 2008.

[18] M. Brand, Charting a manifold, Tech. Rep. 15, Mitsubishi Electric Research Laboratories (MERL), URL `http://www.merl.com/publications/TR2003-013/`, 2003.

[19] J. B. Tenenbaum, V. d. Silva, J. C. Langford, A Global Geometric Framework for Nonlinear Dimensionality Reduction, Science 290 (5500) (2000) 2319–2323, doi:10.1126/science.290.5500.2319, URL `http://www.sciencemag.org/cgi/content/abstract/290/5500/2319`.

[20] S. T. Roweis, L. K. Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, Science 290 (5500) (2000) 2323–2326, doi:10.1126/science.290.5500.2323, URL `http://www.sciencemag.org/cgi/content/abstract/290/5500/2323`.

[21] A. Wismüller, The exploration machine: a novel method for analyzing high-dimensional data in computer-aided diagnosis, in: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 7260 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, 72600G–72600G–7, doi:10.1117/12.813892, 2009.

[22] G. Hinton, S. Roweis, Stochastic Neighbor Embedding, in: Advances in Neural Information Processing Systems 15, MIT Press, 833–840, 2003.

[23] T. Liu, A. Moore, A. Gray, K. Yang, An investigation of practical approximate nearest neighbor algorithms, Advances in neural information processing systems .

[24] A. Moore, An intoductory tutorial on kd-trees, Ph.D. thesis, University of Cambridge, technical Report No. 209, 1990.

[25] A. Gray, A. Moore, N-Body'problems in statistical learning, Advances in Neural Information Processing Systems (2001) 521–527.

[26] T. Kohonen, Self-Organizing Maps, Springer, Berlin, Heidelberg, New York, 3rd edn., 2001.

[27] B. Hammer, M. Strickert, T. Villmann, On the Generalization Ability of GRLVQ Networks, Neural Processing Letters 21 (2) (2005) 109–120.

[28] B. Hammer, T. Villmann, Generalized relevance learning vector quantization, Neural Networks 15 (8-9) (2002) 1059–1068.

[29] P. Schneider, M. Biehl, B. Hammer, Relevance Matrices in LVQ, in: M. Verleysen (Ed.), Proc. of European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, 37–42, 2007.

[30] K. Q. Weinberger, J. Blitzer, L. K. Saul, Distance Metric Learning for Large Margin Nearest Neighbor Classification, Advances in Neural Information Processing Systems 18 (2006) 1473–1480.

[31] K. Bunte, M. Biehl, N. Petkov, M. F. Jonkman, Adaptive Metrics for Content Based Image Retrieval in Dermatology, in: M. Verleysen (Ed.), Proc. of European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, 129–134, 2009.

[32] A. S. Sato, K. Yamada, Generalized learning vector quantization, in: Advances in Neural Information Processing Systems, vol. 8, 423–429, 1996.

[33] P. Schneider, M. Biehl, B. Hammer, Adaptive relevance matrices in Learning Vector Quantization, Neural Computation (2009) In Press.

[34] P. Schneider, K. Bunte, B. Hammer, T. Villmann, M. Biehl, Regularization in matrix relevance learning, Tech. Rep. MLR-02-2008, Leipzig University, iSSN:1865-3960 http://www.uni-leipzig.de/∼compint/mlr/mlr_02_2008.pdf, 2008.

[35] M. Koeber, U. Schäfer, The unique square root of a positive semidefinite matrix, International Journal of Mathematical Education in Science and Technology 37 (8) (2006) 990 – 992, URL `http://www.informaworld.com/10.1080/00207390500285867`.

[36] Z. Zhang, H. Zha, Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment, SIAM Journal of Scientific Computing 26 (2002) 313–338.

[37] L. van der Maaten, G. Hinton, Visualizing Data using t-SNE, Journal of Machine Learning Research 9 (2008) 2579–2605, URL `http://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf`.

[38] K. Weinberger, L. Saul, An introduction to nonlinear dimensionality reduction by maximum variance unfolding, in: Proceedings of the National Conference on Artificial Intelligence, vol. 21, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 1683, 2006.

[39] H. Wang, J. Zheng, Z. an Yao, L. Li, Improved Locally Linear Embedding Through New Distance Computing, in: J. Wang, Z. Yi, J. M. Zurada, B.-L. Lu, H. Yin (Eds.), ISNN (1), vol. 3971 of *Lecture Notes in Computer Science*, Springer, ISBN 3-540-34439-X, 1326–1333, 2006.

[40] L. Zhao, Z. Zhang, Supervised locally linear embedding with probability-based distance for classification, Comput. Math. Appl. 57 (6) (2009) 919–926, ISSN 0898-1221, doi:http://dx.doi.org/10.1016/j.camwa.2008.10.055.

[41] L. K. Saul, S. T. Roweis, Think Globally, Fit Locally: Unsupervised Learning of Nonlinear Manifolds, Journal of Machine Learning Research 4 (2003) 119–155.

[42] A. Wismüller, The Exploration Machine – A novel method for structure-preserving dimensionality reduction, in: M. Verleysen (Ed.), European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, 59–64, 2009.

[43] A. Wismüller, A computational framework for exploratory data analysis, in: M. Verleysen (Ed.), European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, 547–552, 2009.

[44] A. Wismüller, Exploratory Morphogenesis (XOM): A Novel Computational Framework for Self-Organization, Ph.D. thesis, Technical University of Munich, Department of Electrical and Computer Engineering, 2006.

[45] A. Wismüller, A computational framework for exploratory data analysis in biomedical imaging, in: Proceedings of SPIE, vol. 7260, doi:10.1117/12.813892, 2009.

[46] S. Aeberhard, D. Coomans, O. de Vel, Comparison of classifiers in high dimensional settings, Tech. Rep. 02, James Cook University, 1992.

[47] A. Asuncion, D. J. Newman, S. Hettich, C. L. Blake, C. J. Merz, UCI Repository of machine learning databases, `http://archive.ics.uci.edu/ml/`, last visit 19.06.2009, 1998.

[48] S. Rogers, M. Girolami, Multi-class semisupervised learning with the e-truncated multinomial probit Gaussian process, in: Journal of Machine Learning Research: Gaussian Processes in Practice, 1, 1732, 2007.

[49] E. Levina, P. J. Bickel, Maximum Likelihood Estimation of Intrinsic Dimension, in: In Advances in NIPS, MIT Press, Cambridge, USA, 777–784, 2005.