Dimensionality reduction mappings

Kerstin Bunte, Michael Biehl
University of Groningen
Johann Bernoulli Institute for Mathematics
and Computer Science
Nijenborgh 9, Groningen - The Netherlands
Email: k.bunte@rug.nl

Barbara Hammer
University of Bielefeld,
CITEC Center of Excellence
Bielefeld - Germany
Email: bhammer@techfak.uni-bielefeld.de

***Abstract*—A wealth of powerful dimensionality reduction methods has been established which can be used for data visualization and preprocessing. These are accompanied by formal evaluation schemes, which allow a quantitative evaluation along general principles and which even lead to further visualization schemes based on these objectives. Most methods, however, provide a mapping of a priorly given finite set of points only, requiring additional steps for out-of-sample extensions. We propose a general view on dimensionality reduction based on the concept of cost functions, and, based on this general principle, extend dimensionality reduction to explicit mappings of the data manifold. This offers simple out-of-sample extensions. Further, it opens a way towards a theory of data visualization taking the perspective of its generalization ability to new data points. We demonstrate the approach based on a simple global linear mapping as well as prototype-based local linear mappings.**

I. Introduction

The amount of electronic data available today doubles approximately every 20 months. At the same time, its complexity and dimensionality increases dramatically due to improved sensor technology, dedicated data formats, and rapidly increasing capabilities to digitally capture different data modalities. As a consequence, data can no longer be inspected manually, rather, automated methods which help humans to quickly scan through massive data volumes are needed. Data visualization relies on the astonishing cognitive capabilities of humans for structure detection in visual images. In this context, the available information and structural characteristics or specifics can be captured almost instantly by humans despite the given number of data points which are represented in the visualization. As a consequence, data visualization and dimensionality reduction play a key role in modern data mining techniques.

A plethora of methods for dimensionality reduction has been proposed in the past years, see e.g. [1], [2], [3], [4], [5]. In general, the task is to substitute data points in a high dimensional data manifold by lower dimensions (ideally two dimensions to obtain a visualization), such that as much information as possible is preserved. Since this problem formulation is ill-posed, a variety of methods can be derived by imposing additional constraints on the visualization task. Spectral dimensionality reduction techniques such as LLE [6], Isomap [7], or Laplacian eigenmaps [8] rely on the spectrum of the neighborhood graph of the data and preserve important properties of this graph. In general, they allow a unique algebraic solution of the corresponding mathematical objective which formalizes the visualization task. Thereby, many methods rely on very simple affinity functions such as Gaussians such that their results are flawed when it comes to boundaries or separated manifolds. Using more complex affinities such as present in Isomap [7] or maximum variance unfolding [9] can partially avoid this problem at the prize of higher computational costs. Nonlinear methods often have the drawback that local optima can easily occur. Their results can be more appropriate as demonstrated e.g. in [10], [3], [11].

All of these methods, however, map the given data points only and their extension towards novel data points requires additional effort. Essentially, two different ways for out of sample extensions can be found in the literature: either an interpolation takes place, e.g. by fitting a neural network to the data which interpolates the projection mapping. This has the drawback that the mapping is not optimized for the projection task, rather, it interpolates the given (probably faulty) coordinates. Alternatively, novel points can be directly mapped to a position in the projection space which minimizes the underlying cost function of the visualization method, where the coordinates of the priorly given data and their projections are kept fixed. In some cases, an explicit algebraic expression is possible, for complex cost functions, numerical optimization is necessary. Usually, however, the novel coordinates depend on all given data by means of the cost function, which often yields to quadratic effort corresponding to the pairwise affinities of data points captured in the cost function.

In this contribution we propose a general principle how dimensionality reduction mappings which are optimized for the visualization task can be obtained based on the dimensionality reduction principles as proposed in the literature. For this purpose, a specific form and complexity of the dimensionality reduction mapping is fixed, such as a function stemming from a class which allows universal approximation, e.g. locally linear functions, or a particularly simple function to allow easy interpretability such as a global linear function. Instead of the coordinates of the projected data points, the function parameters are optimized in a second step. A similar mechanism has been proposed in specific settings in the contribution [6], LLE is extended towards a locally linear embedding function, leading to locally linear coordination, in the approach [12] t-SNE is extended towards an embedding given by an encoder networks. We argue that this principle can be generalized to a general framework which allows to adapt embedding functions of different complexity according to a given objective induced by a dimensionality reduction technologies. We exemplarily demonstrate this procedure for global linear mappings and local linear mappings built on top of prototype based methods, and the visualization cost term of t-SNE. For both cases, visualization mappings can be inferred which can be described by only few model parameters.

The fact that an explicit mapping is obtained instead of coordinates of single points has several benefits: out-of-sample extensions are immediate and reduce to (efficient) function evaluations, whereby the form and complexity of the function can be defined a priori. Approximate inverse mappings can be constructed e.g. by a local linear approximation of the projection and the corresponding pseudoinverse. This way, paths in the projection space can be traced back to paths in the data manifold, shedding some light on the structure of the projection. Since the dimensionality reduction mapping is usually described by a small number of parameters, few data points are sufficient to reliably determine these parameters, i.e. training can be done using a small subset of the data only instead of the full data set. This can dramatically reduce the complexity of the computation since the cost functions often scale at least quadratically with the number of training data. This generalization ability of dimensionality reduction mapping can formally be put into the framework of statistical learning theory. Assuming that a loss function of the dimensionality reduction is fixed, the empirical error of this loss function on a small data set is often already representative for the full error assumed reasonable mappings and loss functions are considered. We will discuss this fact in more detail within this contribution. Further, we will also discuss, in how far this generalization ability can be used to show a formal concept of learnability of dimensionality reduction e.g. based on the reconstruction error of the map.

II. Dimensionality reduction as cost optimization

First, we shortly review some of the most popular dimensionality reduction methods as proposed in the literature. We assume that high dimensional points $X : \{\vec{x}^i \in \mathbb{R}^D\}_{i=1}^n$ are given which should be projected to points $Y : \{\vec{y}^i \in \mathbb{R}^d\}_{i=1}^n$ with $d < D$, usually $d = 2$ for visualization. Corresponding distances are denoted as $d_{\mathcal{X}}(\vec{x}^i, \vec{x}^j)$ for the original manifold, and $d_{\mathcal{E}}(\vec{y}^i, \vec{y}^j)$ for the projection space. Usually, $d_{\mathcal{E}}$ is chosen as the Euclidean distance, while $d_{\mathcal{X}}(\vec{x}^i, \vec{x}^j)$ can be picked arbitrarily (e.g according to Euclidean or geodesic distances in the high dimensional space.)

Multidimensional scaling and extensions: Multidimensional scaling (MDS) [13] constitutes probably one of the oldest dimensionality reduction methods. Its goal is to find projections such that the pairwise relations of data are preserved as much as possible as measured in the least squares sense, i.e.

$$E_{\mathrm{MDS}} = \sum_{ij} ((\vec{x}^i)^\top \vec{x}^j - (\vec{y}^i)^\top \vec{y}^j)^2$$

is minimized where, for original MDS, the pairwise relation of data is measured in terms of dot products in the original or projection space, respectively. This formulation has the benefit that an analytical solution is possible in terms of the eigenvectors of the Gram matrix. This objective has later been generalized to explicitly preserve distances:

$$E_{\mathrm{MDS}} = \frac{1}{c} \sum_{ij} w_{ij} (d_{\mathcal{X}}(\vec{x}^i, \vec{x}^j) - d_{\mathcal{E}}(\vec{y}^i, \vec{y}^j))^2$$

with Euclidean distances, where the weights w_{ij} can be chosen appropriately, e.g. $w_{ij} = 1$, and c is a normalizing constant [1]. For the popular Sammon mapping, the weights are picked as $w_{ij} = 1/d_{\mathcal{X}}(\vec{x}^i, \vec{x}^j)$, this way putting most emphasis on the preservation of small distances, and c denotes the sum over these distances. In this case, optimization of the cost function usually takes place by means of a gradient descent.

Isomap: Isomap [7] is based on the observation that the Euclidean distance is often not appropriate to describe pairwise relations of data , rather, the distance should be measured along the data manifold. Therefore, Isomap is based on an approximation of the manifold distance by geodesic distances, i.e. shortest paths lengths in the graph which results if every data point is connected to its nearest neighbors (using either k-neighborhoods or ϵ-balls to define the local neighborhood).

Locally linear embedding: Locally linear embedding (LLE) [6] first expresses local topologies by reconstructing a data point by linear combinations of its local neighborhood (denoted by $i \to j$) in the original space under the constraint that the coefficients sum to one such that translation and rotation invariance is enforced: minimize $\sum_i (\vec{x}^i - \sum_{i \to j} w_{ij} \vec{x}^j)^2$ with $\sum w_{ij} = 1$. Afterwards, projections are determined such that the local linear relationships are preserved as much as possible in a least squares sense where a normalization of the coefficients leads to a unique optimum: minimize $\sum_i (\vec{y}^i - \sum_{i \to j} w_{ij} \vec{y}^j)^2$ such that $\sum \vec{y}^i = 0$ and $\mathbf{Y}^t \mathbf{Y} = \mathbf{n}$, the latter referring to the corresponding matrices.

Laplacian Eigenmaps: Laplacian eigenmaps [8], like LLE and Isomap, start with a local neighborhood graph given by the k nearest neighbors or ϵ-neighborhood, respectively. The connections are weighted with values w_{ij}, e.g. using the heat kernel. Then, projection takes place by picking the eigendirections corresponding to the smallest eigenvalues larger than 0 as computed in the generalized eigenvalue problem given by the corresponding graph Laplacian and the degree matrix of the graph. This is equivalent to minimizing the embedding objective $\sum_{i \to j} w_{ij} d_{\mathcal{E}}(\vec{y}^i, \vec{y}^j)^2$ with Euclidean distance, under the constraint $\mathbf{Y}^t D \mathbf{Y} = \mathbf{1}$ and $\mathbf{Y}^t D \vec{1} = \vec{0}$, where D is the degree matrix and $\mathbf{Y}$ refers to the matrix of coefficients, to remove scaling factors and translation factors.

Maximum variance unfolding: Maximum variance unfolding (MVU) [9] also first determines a neighborhood graph by taking the k nearest neighbors or ϵ neighborhoods. Afterwards, it finds projections $\vec{y}^i$ such that the variance of the projection is maximized, i.e. $\sum_{ij} d_{\mathcal{E}}(\vec{y}^i, \vec{y}^j)^2$ is maximum subject to a preservation of neighbors, i.e. $d_{\mathcal{E}}(\vec{y}^i, \vec{y}^j) = d_{\mathcal{X}}(\vec{x}^i, \vec{x}^j)$ for all neighbored points $\vec{x}^i$ and $\vec{x}^j$, and the normalization $\sum \vec{y}^i = 0$. This can be reformulated as a convex problem by considering the variables $(\vec{y}^i)^\top \vec{y}$ instead. Further, it is not clear that a solution exists due to the constraints, such that possibly slack variables have to be introduced.

Stochastic neighbor embedding: Stochastic neighbor embedding (SNE) [10] defines probabilities

$$p_{j|i} = \frac{\exp\left(\frac{-d_{\mathcal{X}}(\vec{x}^i, \vec{x}^j)^2}{2\sigma_i}\right)}{\sum_{k \neq i} \exp\left(\frac{-d_{\mathcal{X}}(\vec{x}^i, \vec{x}^k)^2}{2\sigma_i}\right)}$$

and

$$q_{j|i} = \frac{\exp\left(-d_{\mathcal{E}}(\vec{y}^i, \vec{y}^j\right)^2}{\sum_{k\neq i}\exp\left(-d_{\mathcal{E}}(\vec{y}^i, \vec{y}^k)^2\right)}$$

with Euclidean distances as default. The goal is to optimize the Kullback-Leibler divergence $E_{\mathrm{SNE}} = -\sum_{ij} p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$, where bandwidths σ_i are determined based on the so-called perplexity which determines the number of neighbors of a given point. A gradient descent is used for the optimization.

T-distributed stochastic neighbor embedding: t-distributed SNE (t-SNE) [3] slightly modifies the SNE cost function and uses a distribution in the embedding space with long tails, student-t. Its cost function is

$$E_{\mathrm{t-SNE}} = \sum_i \sum_j p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right)$$

where

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

symmetrizes the conditional probabilities, n denoting the number of data points, and

$$q_{ij} = \frac{(1 + d_{\mathcal{E}}(\vec{y}^i, \vec{y}^j)/\varsigma)^{-\frac{\varsigma+1}{2}}}{\sum_{k\neq l}(1 + d_{\mathcal{E}}(\vec{y}^k, \vec{y}^l)/\varsigma)^{-\frac{\varsigma+1}{2}}}$$

is given by student-t with parameter $\varsigma = -1$, for example. Optimization takes place by means of a gradient method.

A general view

These methods obey one general principle: characteristics of the data $\vec{x}$ are computed and projections $\vec{y}$ are determined such that the corresponding characteristics of the projections are as close to the characteristics of $\vec{x}$ as possible, fulfilling possibly additional constraints or objectives to achieve uniqueness. Thereby, the methods differ in the way how data characteristics are determined and how exactly the similarity of the characteristics is defined and optimized. Table I summarizes the properties of the optimization methods under this point of view. Naturally, the methods severely differ with respect to the way in which optimization takes place: in some cases, the characteristics can be directly computed from the data (such as distances), in others, an optimization step is required (such as local linear weights). In some cases, the optimization of the error measure can be done in closed form (such as for Laplacian eigenmaps), in other cases, numerical optimization is necessary (such as for t-SNE).

III. Dimensionality reduction mapping

All dimensionality reduction methods as introduced above give a mapping of points only: $\vec{x}^i \mapsto \vec{y}^i$. Extensions of the map to new data points $\vec{x}$ require a new computation, often the respective coefficients which minimize the objective of dimensionality reduction are determined, keeping all known coefficients fixed. This method has the drawback that additional effort is required if new data points are dealt with. Further, it is not easily possible to formalize and investigate the generalization ability of these mappings, i.e. the question, whether the method works well for future data from the same manifold assumed it works well for the known training set.

These issues can be circumvented if a dimensionality reduction mapping

$$f : \mathcal{X} \to \mathcal{E}, \vec{x}^i \mapsto \vec{y}^i = f(\vec{x}^i)$$

from the space $\mathcal{X}$ of original data X to the embedding space $\mathcal{E}$ of the projected points Y is computed rather than single coefficients $\vec{x}^i \mapsto \vec{y}^i$ only.

Previous work

In the literature, a few dimensionality reduction technologies provide an explicit mapping of the data: linear methods such as PCA provide an explicit linear function which optimizes the information loss while projecting [14]. Extensions to nonlinear functions are given by autoencoder networks, which provide a function given by a multilayer feedforward network in such a way that the reconstruction error is minimized when back projecting with another feedforward network [2]. Typically, training takes place by standard back propagation directly minimizing the reconstruction error. Manifold charting starts from locally linear embeddings given by local PCAs and glues these pieces together by minimizing the error on the overlaps [15], [16]. This way, a global embedding mapping is obtained. Topographic maps such as the self-organizing map or generative topographic mapping characterize data in terms of prototypes which are visualized in low dimensions [17], [18]. Due to the clustering, new data can directly be visualized by mapping these data to their closest prototype or its visualization, respectively.

A few dimensionality reduction mappings which give coordinates per default as introduced above have been extended to global dimensionality reduction mappings. Locally linear coordination (LLC) extends LLE in the following way [19]: it is assumed that local linear dimensionality reduction methods are available, such as local PCAs. These are glued together adding affine transformations. These additional parameters are optimized by inserting the resulting points in the LLE cost function and corresponding optimization. Kernel maps, based on the ideas of kernel eigenmap methods, provide out-of-sample extensions [20]. And parameterized t-SNE [12] extends t-SNE towards an embedding given by a multilayer neural network. The network parameters are determined using back propagation, where, instead of the mean squared error, the t-SNE cost function is taken as objective.

A general principle

Considering dimensionality reduction as optimization task as formulated in Table I allows to simultaneously extend all methods to dimensionality reduction mappings a general way. In a first step, the principled form and complexity of the dimensionality reduction mapping is fixed: a parameterized function

$$f_W : \mathcal{X} \to \mathcal{E}$$

is chosen with parameters W which have to be determined such that the projections are satisfactory. The form of this function can be given by a linear function, a locally linear function, a feedforward neural network, etc. Then, instead of coefficients $\vec{y}^i$, the images of the map $f_W(\vec{x}^i)$ are considered and instead of the single coefficients, the map parameters W

method	characteristics of data	characteristics of projections	error measure
MDS	Euclidean distance $d_{\mathcal{X}}(\vec{x}^i, \vec{x}^j)$	Euclidean distance $d_{\mathcal{E}}(\vec{y}^i, \vec{y}^j)$	minimize weighted least squared error
Isomap	Geodesic distance $d_{\text{geodesic}}(\vec{x}^i, \vec{x}^j)$	Euclidean distance $d_{\mathcal{E}}(\vec{y}^i, \vec{y}^j)$	minimize weighted least squared error
LLE	reconstruction weights w_{ij} such that $\sum(\vec{x}^i - \sum_{i \to j} w_{ij}\vec{x}^j)^2$ is minimum with constraints $\sum_j w_{ij} = 1$	reconstruction weights $\tilde{w}_{ij}$ such that $\sum(\vec{y}^i - \sum_{i \to j} \tilde{w}_{ij}\vec{y}^j)^2$ is minimum with constraints $\sum \vec{y}^i = 0$, $\mathbf{Y}^t\mathbf{Y} = \mathbf{n}$	enforce identity $w_{ij} = \tilde{w}_{ij}$
Laplacian eigenmap	negative heat kernel weights $-w_{ij} = \exp(-d_{\mathcal{X}}(\vec{x}^i, \vec{x}^j)^2/t)$ for $i \to j$	squared Euclidean distance $d_{\mathcal{E}}(\vec{y}^i, \vec{y}^j)^2$ for $i \to j$ with constraints $\mathbf{Y}^t D\mathbf{Y} = \mathbf{1}$, $\mathbf{Y}^t D\vec{1} = \vec{0}$	maximize correlation
MVU	Euclidean distance $d_{\mathcal{X}}(\vec{x}^i, \vec{x}^j)$ for $i \to j$	Euclidean distance $d_{\mathcal{E}}(\vec{y}^i, \vec{y}^j)$ for $i \to j$ such that $\sum_{ij} d_{\mathcal{E}}(\vec{y}^i, \vec{y}^j)^2$ is maximum and $\sum_i \vec{y}^i = 0$.	enforce identitiy (introducing slack variables if necessary)
SNE	probabilities $p_{j\|i} = \frac{\exp\left(-d_{\mathcal{X}}(\vec{x}^i,\vec{x}^j)^2/2\sigma_i\right)}{\sum_{k \neq i} \exp\left(-d_{\mathcal{X}}(\vec{x}^i,\vec{x}^k)^2/2\sigma_i\right)}$	probabilities $q_{j\|i} = \frac{\exp\left(-d_{\mathcal{E}}(\vec{y}^i,\vec{y}^j)\right)^2}{\sum_{k \neq i} \exp\left(-d_{\mathcal{E}}(\vec{y}^i,\vec{y}^k)^2\right)}$	minimize Kullback-Leibler divergences
t-SNE	probabilities $p_{ij} = \frac{p_{j\|i} + p_{i\|j}}{2n}$	probabilities $q_{ij} = \frac{(1+d_{\mathcal{E}}(\vec{y}^i,\vec{y}^j)/\varsigma)^{-\frac{\varsigma+1}{2}}}{\sum_{k \neq l}(1+d_{\mathcal{E}}(\vec{y}^k,\vec{y}^l)/\varsigma)^{-\frac{\varsigma+1}{2}}}$	minimize Kullback-Leibler divergence

TABLE I

MANY DIMENSIONALITY REDUCTION METHODS CAN BE PUT INTO A GENERAL FRAMEWORK: CHARACTERISTICS OF THE DATA ARE EXTRACTED. PROJECTIONS LEAD TO CORRESPONDING CHARACTERISTICS DEPENDING ON THE COEFFICIENTS. THESE COEFFICIENTS ARE DETERMINED SUCH THAT AN ERROR MEASURE OF THE CHARACTERISTICS IS MINIMIZED, FULFILLING PROBABLY ADDITIONAL CONSTRAINTS.

are optimized. For this purpose characteristics of the data $\vec{x}^i$ can be computed as before. Characteristics of the projected points depend on the parameterized quantities $f_W(\vec{x}^i)$ instead of the coefficients. These terms can be plugged into the corresponding error measure and the parameters W can be determined via optimization taking the same constraints into account as before (or relaxations thereof).

This principle leads to a well defined mathematical objective for the mapping parameters W for every dimensionality reduction method as summarized above, although the way in which optimization takes place is possibly different as compared to the original method: while numerical methods such as gradient descent can still be used, it is probably no longer possible to find closed form solutions for spectral methods. However, numerical optimization can be used as a default in all cases.

We exemplarily derive formulas for two specific cases: a global linear mapping and local linear mappings built on top of local linear projections, whereby we combine these functions with the t-SNE cost term in both cases. The suitability of the general principle for different dimensionality reduction cost functions and different parameterizations of the projection mapping will be the subject of future work.

Linear t-SNE Mapping

We derive the formulation in case of a linear hypothesis for the mapping of the high-dimensional data points $\vec{x}^l$ and the t-SNE cost function. The mapping f_W becomes

$$f_W : \vec{x}^l \to \vec{y}^l = A \cdot \vec{x}^l \quad .$$

The rectangular matrix A defines a linear mapping from $\mathbb{R}^D \to \mathbb{R}^d$. This matrix can be optimized using a stochastic gradient descent procedure using the following gradient of the t-SNE cost function:

$$\begin{aligned}\frac{\partial E_{\text{t-SNE}}}{\partial A} &= \sum_i \sum_j \frac{\partial E_{\text{t-SNE}}}{\partial q_{ij}} \cdot \frac{\partial q_{ij}}{\partial d_{\mathcal{E}}(\vec{y}^i, \vec{y}^j)^2} \cdot \frac{\partial d_{\mathcal{E}}(\vec{y}^i, \vec{y}^j)^2}{\partial A} \\ &= \frac{\varsigma + 1}{2\varsigma} \sum_i \sum_j (p_{ij} - q_{ji}) \cdot \\ &\qquad (1 + d_{\mathcal{E}}(\vec{y}^i, \vec{y}^j)/\varsigma)^{-1} \cdot \frac{\partial d_{\mathcal{E}}(\vec{y}^i, \vec{y}^j)^2}{\partial A}\end{aligned}$$

with Euclidean distance $d_{\mathcal{E}}(\vec{y}^i, \vec{y}^j) = ||A\vec{x}^i - A\vec{x}^j||$ follows:

$$\frac{\partial d_{\mathcal{E}}(\vec{y}^i, \vec{y}^j)^2}{\partial A} = 2(A\vec{x}^i - A\vec{x}^j)(\vec{x}^i - \vec{x}^j)$$

Hence

$$\begin{aligned}\frac{\partial E_{\text{t-SNE}}}{\partial A} &= \frac{\varsigma + 1}{\varsigma} \sum_i \sum_j (A\vec{x}^i - A\vec{x}^j)(\vec{x}^i - \vec{x}^j) \cdot \\ &\qquad (p_{ij} - q_{ji}) \frac{1}{1 + ||A\vec{x}^i - A\vec{x}^j||^2/\varsigma} \quad .\end{aligned}$$

We test this procedure in comparison to simple PCA on a three dimensional benchmark: three Gaussians are stacked together as shown in Fig. 1. Because of the large variance in the z-direction, a PCA mapping projects the data clouds onto each other. In contrast, a linear mapping trained such that the t-SNE cost function of the projections is optimized leads to a much clearer separation of the cluster structure, because it takes into account the preservation of local structures as measured by the t-SNE cost function. Fig. 1 clearly shows the superiority of the mapping obtained this way, referred to as DiReduct mapping. In addition, the projection is formally evaluated using the error measure as proposed in [21], [22]. Roughly speaking, these rely on the k-intrusions and k-extrusions in the projections, i.e. k-nearest neighbors in the projection, but not the original space, and vice versa. The quality measures refer to the quantities Q which measures

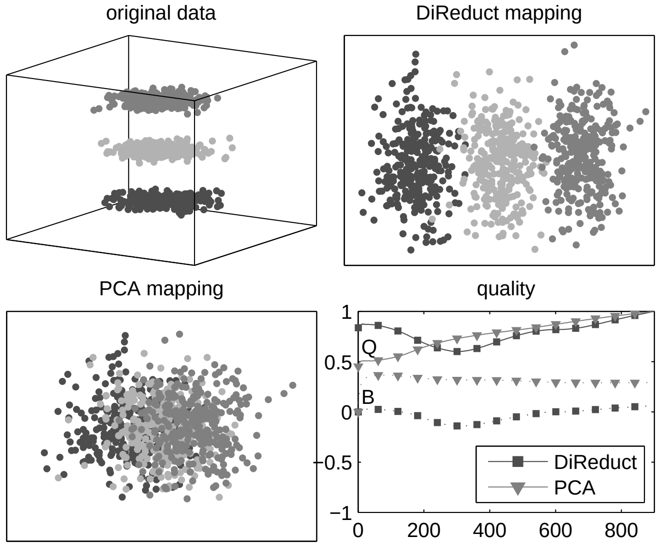

Fig. 1. Simulation Results for a globally linear map trained with PCA and an optimization of the t-SNE costs, respectively. The latter leads to a better separation due to its local nature, which can be formally evaluated referring to the intrusions and extrusions of the mapping.

the percentage of data which is not k-intrusive or k-extrusive, and B which measures the percentage of k-intrusions minus the percentage of k-extrusions of the map, i.e. it characterizes the behavior of the mapping. Obviously, DiReduct shows a superior quality in particular for small neighborhood ranges since it better preserves local structures of the data. Further, unlike PCA which displays a trend towards intrusions, it is rather neutral in the mapping character, being mildly extrusive for medium sizes of k nearest neighbors.

Locally linear t-SNE mapping

As an alternative, we can built a locally linear embedding function on top of locally linear projections obtained e.g. using prototype based methods such as neural gas with local PCA, mixture of probabilistic PCA, or even supervised clustering such as learning vector quantization with adaptive matrices [23], [24]. We assume that locally linear projections of the data points are derived from these techniques:

$$\vec{x}^l \mapsto p_k(\vec{x}^l) = \Omega_k \vec{x}^l - \vec{w}^k$$

with local matrices Ω_k and offsets $\vec{w}^k$. Further, we assume the existence of responsibilities r_{lk} of mapping p_k for $\vec{x}^l$, which can be given by the receptive fields of the locally linear maps centered around $\vec{w}^k$ or Gaussians centered around these points, for example. We assume $\sum_k r_{lk} = 1$. Then a global mapping which combines these linear pieces can be defined as

$$f_W : \vec{x}^l \mapsto \vec{y}^l = \sum_k r_{lk}(L_k \cdot p_k(\vec{x}^l) + l_k) \quad ,$$

using local linear projections L_k and local offsets l_k to align the local pieces. Note that the dimensionality of the weights W which have to be determined depends on the number of pieces k and the dimensionality of the local projections. Usually, it is much smaller than the number of coefficients when projecting all points $\vec{y}^l$ directly to the Euclidean plane.

These parameters can be determined by a stochastic gradient descent. The derivative of the t-SNE cost function yields

$$\begin{aligned}\frac{\partial E_{\mathrm{t-SNE}}}{\partial L_k} =& \sum_{ij} \frac{\partial E_{\mathrm{t-SNE}}}{\partial q_{ij}} \cdot \frac{\partial q_{ij}}{\partial d_{\mathcal{E}}(\vec{y}^i,\vec{y}^j)^2} \cdot \frac{\partial d_{\mathcal{E}}(\vec{y}^i,\vec{y}^j)^2}{\partial L_k}\\ =& \frac{\varsigma+1}{2\varsigma}\sum_{ij}(p_{ij}-q_{ji})\frac{1}{1+d_{\mathcal{E}}(\vec{y}^i,\vec{y}^j)^2/\varsigma}\cdot\\ & \frac{\partial d_{\mathcal{E}}(\vec{y}^i,\vec{y}^j)^2}{\partial L_k}\\ =& \frac{\varsigma+1}{\varsigma}\sum_{ij}(p_{ij}-q_{ji})\frac{1}{1+d_{\mathcal{E}}(\vec{y}^i,\vec{y}^j)/\varsigma}\cdot\\ & (\vec{y}^i-\vec{y}^j)(r_{ik}p_k(\vec{x}^i)-r_{jk}p_k(\vec{x}^j))\end{aligned}$$

and

$$\begin{aligned}\frac{\partial E_{\mathrm{t-SNE}}}{\partial l_k} =& \sum_{ij} \frac{\partial E_{\mathrm{t-SNE}}}{\partial q_{ij}} \cdot \frac{\partial q_{ij}}{\partial d_{\mathcal{E}}(\vec{y}^i,\vec{y}^j)^2} \cdot \frac{\partial d_{\mathcal{E}}(\vec{y}^i,\vec{y}^j)^2}{\partial l_k}\\ =& \frac{\varsigma+1}{\varsigma}\sum_{ij}(p_{ij}-q_{ji})\frac{1}{1+d_{\mathcal{E}}(\vec{y}^i,\vec{y}^j)^2/\varsigma}\cdot\\ & (\vec{y}^i-\vec{y}^j)(r_{ik}-r_{jk}) \quad .\end{aligned}$$

assuming Euclidean distance in the projection space, as before.

We demonstrate the suitability of this approach in two settings. We consider the classical USPS data set consisting of 11000 samples of handwritten digits, comprising 10 classes according to the digits, every digit represented by 16×16 grey values. We test two settings: in the first setting, local linear maps $p_k(\vec{x}^i)$ are obtained by an unsupervised prototype-based clustering of the data set; the responsibilities r_{lk} are given by the receptive fields. The local linear maps p_k consist of an offset given by the prototypes, i.e. cluster centers, and local PCA projections in the receptive fields into the main eigendirections which are directly determined based on the receptive fields. We choose 20 clusters for the clustering algorithm and dimensionality 30 for the PCA projection. The clustering is obtained using batch neural gas as a very robust and fast clustering algorithm with few parameters (the number of epochs is chosen as 30, the neighborhood cooperation is multiplicatively annealed from 10 to almost 0) [25].

On top of these local linear projections, linear transformations are adapted such that the local pieces are coordinated. As objective, we use the t-SNE cost function as specified above. Optimization is done by gradient descent with 300 epochs, and learning rate decreasing from 0.5 to 0.1. Initialization of the global mapping takes place by setting the mean of the single projections to 0 and choosing the first two principal directions of the receptive fields as projection.

For clustering and projection, a subsample of size 500 is chosen. An extension to all data is immediate due to the explicit mapping. The result of this procedure is shown in Fig. 2. We report the result of the subsample used for training as well as the extension to the full USPS data set. Interestingly, the generalization is quite good, the overall shape being visible already for the small data set. The nearest neighbor error of the projection for the subsample used for training is 24%, while it is 31% for the full data set.

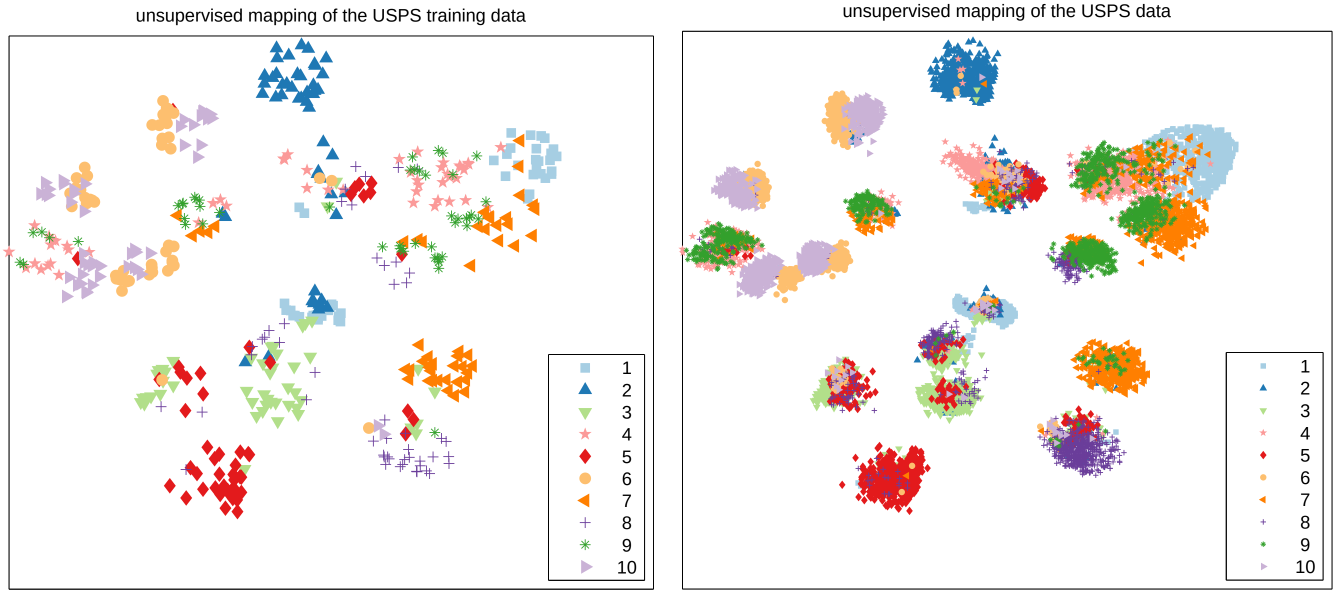

Fig. 2. Projection of a subsample of the USPS data set when combining unsupervised clustering and learning of a mapping. The result of the subsample used for training (left) as well as the full data set (right) is depicted.

The choice of the dimensionality reduction mapping as a composition of locally linear maps and affine coordinations immediately allows to integrate prior knowledge into the visualization. This property is investigated in the following setting: Instead of local linear projections obtained by unsupervised techniques, information obtained by means of supervised learning can be used, such that a strong bias towards this auxiliary information is achieved. We combine locally linear maps as obtained by learning vector quantization (LVQ) and an affine coordination as before. The LVQ training algorithm which we use in this case is given by low rank generalized matrix LVQ as described in [26] since it simultaneously determines receptive fields as well as local matrix projections which improve the classification accuracy as much as possible. We use 10 prototypes for this setting, a projection dimension 2 for the local linear functions, and the default training parameters for the LVQ training procedure. Training of LVQ is done using 200 data points per class. For the coordination, we use only 500 data as before, extending to all data points by means of the trained mapping. The result of this procedure is displayed in Fig. 3. Due to the bias by means of the given class information, the clusters are better separated in this case as compared to the fully unsupervised scenario. As before, the generalization ability of the procedure towards new data is quite good as can be seen in the image. The nearest neighbor classification error decreases only slightly from 11% to 15% for the full data set. Obviously, due to the integration of prior knowledge, the priorly known classes are better captured in the visualized projection compared to the unsupervised setting.

The quality of the projection is evaluated by means of extrusions and intrusions as above, see Fig. 4. The quality of the two mappings is very similar, the unsupervised projection being more reliable for small values of k due to the different focus of the projections. The supervised setting neglects local neighborhood relationships for the sake of a better class structure as characterized by the given auxiliary class labels.

The presented projections have the advantage that an explicit mapping is available. However, the restriction to locally linear functions reduces the flexibility of the mappings as compared to techniques which can freely adapt the coefficients such as original t-SNE. In particular, local nonlinear distorsions cannot be achieved if we restrict to locally linear functions. However, the proposed framework offers a general view on the setting, hence alternative choices are possible and remain to be tested, such as locally nonlinear functions.

IV. Generalization Ability

The extension towards dimensionality reduction mappings offers the possibility to learn the mapping based on few randomly selected data points only. Depending on the size of the data, this can severely improve the performance of the method, since it reduces the squared complexity to a constant effort. However, an assumption underlying this procedure is that the dimensionality reduction mapping generalizes from few data to new data stemming from the same underlying distribution. That means we have to ensure that the quality measure for all data is good assumed it is good for a given finite subsample used to determine the mapping parameters.

Recently, some work on how dimensionality reduction can be formally evaluated has been proposed [22], [5]. As pointed out in [22], one objective of dimensionality reduction is to preserve the available information as much as possible. In consequence, the possibility to reconstruct the points $\vec{x}^i$ from their projections $\vec{y}^i$ can act as valid evaluation measure. Assuming a dimensionality reduction mapping $f : \mathcal{X} \to \mathcal{E}$ is given, this results in the reconstruction error

$$E(P) := \int_{\mathcal{X}} \|\vec{x} - f^{-1}(f(\vec{x}))\|^2 P(\vec{x}) d\vec{x}$$

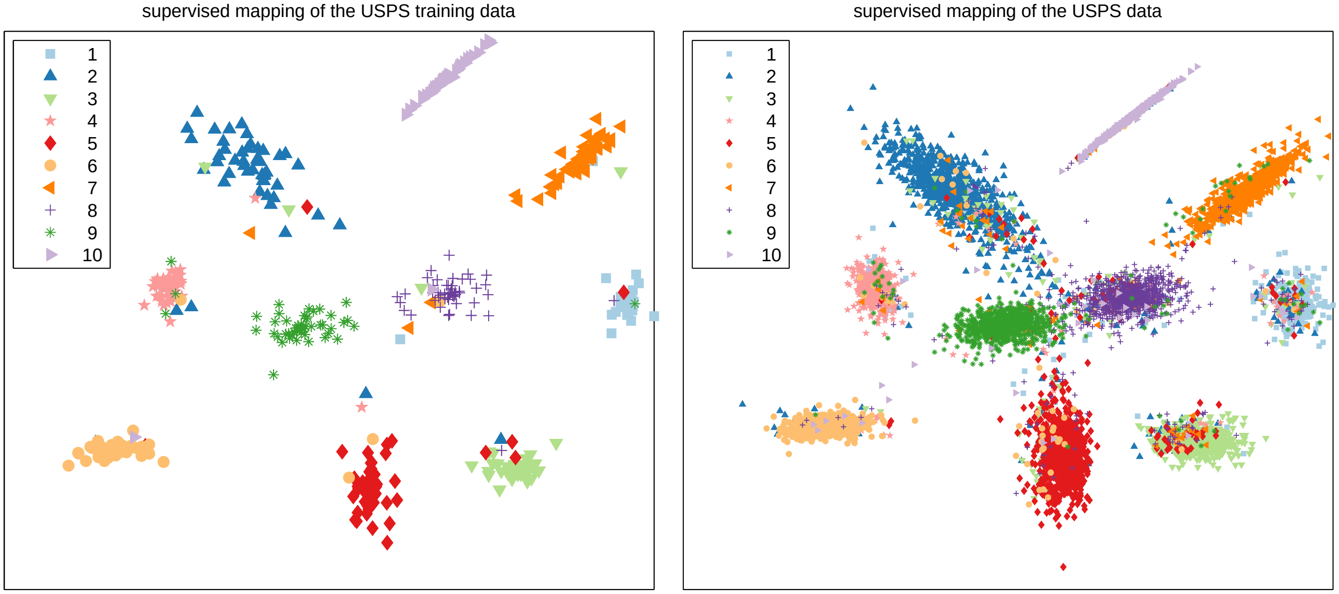

Fig. 3. Projection of a subsample of the USPS data set when combining supervised classification and learning of a mapping. The result of the subsample used for training (left) as well as the full data set (right) is depicted.

where P defines the probability measure according to which the data $\vec{x}$ are distributed in $\mathcal{X}$ and f^{-1} constitutes an approximate inverse mapping of f, an exact inverse in general not existing. Usually, the full data manifold is not available, but a finite set of samples only. Then, the empirical error can be computed

$$\widehat{E}_n(\vec{x}) := \frac{1}{n}\sum_i \|\vec{x}^i - f^{-1}(f((\vec{x}^i)))\|^2$$

for given data $\vec{x}^i$. Note that some dimensionality reduction mappings such as autoencoder networks explicitly optimize this empirical approximation of the costs $E(P)$.

Most dimensionality reduction methods map the data points only, such that neither f nor its approximate inverse are available. Therefore, evaluation measures as proposed in [22], [5] rely on k-neighborhoods in the original and the projection space to approximately capture neighborhood preservation. If a dimensionality reduction mapping is learned, f and its approximate inverse f^{-1} are available. Thus, the evaluation measure $\widehat{E}_n(\vec{x})$ can be evaluated. Since the form of f is fixed prior to training, we can specify a function class $\mathcal{F}$ with $f \in \mathcal{F}$ independently of the given training set. Assuming representative vectors $\vec{x}^i$ are chosen independently and identically distributed according to P the question is whether this quantity allows to limit the real error $E(P)$ we are interested in. As usual, bounds should hold simultaneously for all possible functions in $\mathcal{F}$ to circumvent the problem that the function f is chosen according to the given training data and, thus, the empirical error $\widehat{E}_n(\vec{x})$ is usually small.

This setting can be captured in the classical framework of computational learning theory, as specified e.g. in [27]. We can adapt Theorem 8 from [27] to our setting: We consider a fixed function class

$$\mathcal{F} : \mathcal{X} \to \mathcal{E}$$

from which the dimensionality reduction mapping is taken. We assume without loss of generality, that the norm of the input data and its reconstructions under mappings $f^{-1} \circ f$, f^{-1} denoting the approximate inverse of $f \in \mathcal{F}$, are restricted (scaling the data priorly, if necessary), such that the reconstruction error is induced by the squared error, which is a loss function with limited codomain

$$\mathcal{L} : \mathcal{X} \times \mathcal{X} \to [0,1], (\vec{x}^i, \vec{x}^j) \mapsto \|\vec{x}^i - \vec{x}^j\|^2$$

Then, as reported in [27] (Theorem 8), assuming i.i.d. data according to P, for any confidence $\delta \in (0,1)$ and every $f \in \mathcal{F}$ the following holds

$$E(P) \leq \widehat{E}_n(\vec{x}) + R_n(\mathcal{L}_\mathcal{F}) + \sqrt{\frac{8\ \ln(2/\delta)}{n}}$$

with probability at least $1 - \delta$ where

$$\mathcal{L}_\mathcal{F} := \{\vec{x} \mapsto \mathcal{L}(f^{-1}(f(\vec{x})), \vec{x}) \mid f \in \mathcal{F}\}$$

and R_n refers to the so-called Rademacher complexity of the function class. The Rademacher complexity constitutes a quantity which, similar to the Vapnik Chervonenkis dimension, estimates the capacity of a given function class. Assume σ_i are independent identically distributed $\{\pm 1\}$-valued random variables. The empirical Rademacher complexity of a real valued function class $\mathcal{G}$ is

$$\widehat{R}_n(\mathcal{G}) := \mathbf{E}\left(\sup_{f\in\mathcal{G}}\left|\frac{2}{n}\sum_i \sigma_i f(\vec{x}^i)\right| \quad \text{given } \vec{x}^1, \ldots, \vec{x}^n\right)$$

where the expectation is taken over σ_i. It estimates the expected worst case correlation of functions in $\mathcal{F}$ with random ± 1-valued vectors. The Rademacher complexity denotes the expectation with respect to $\vec{x}$.

This result implies that the generalization ability of dimensionality reduction mappings is usually guaranteed since

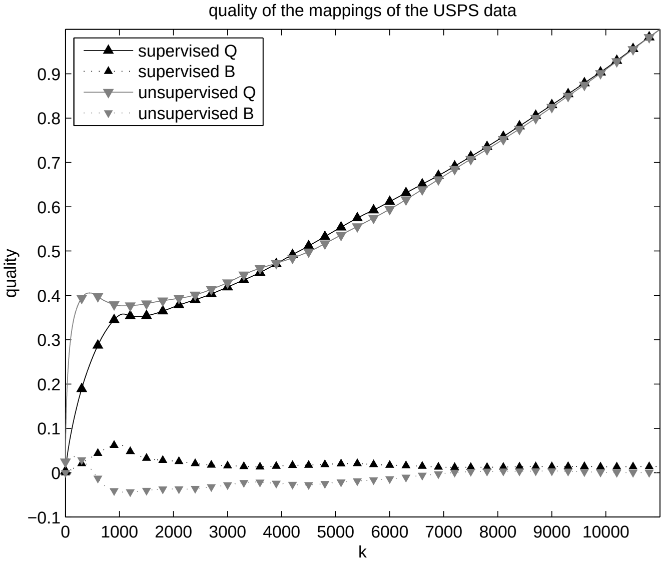

Fig. 4. Quality of the supervised and unsupervised locally linear projection on the USPS data set as measured by the measure as proposed in [22].

the Gaussian complexity of the class $\mathcal{L}_{\mathcal{F}}$ can be limited for reasonable choices of the mapping function $\mathcal{F}$. For linear or piecewise linear functions induced on top of a prototype based tesselation of the data space as considered above, for example, bounds on the Rademacher complexity can be derived in the same way as explained in [27], [24].

V. Conclusion

In this contribution, the question how a dimensionality reduction mapping can be inferred rather than coordinates of separated points has been considered. By formulating dimensionality reduction as an optimization problem of structural characteristics, many classical dimensionality reduction techniques can simultaneously be extended towards explicit mappings which depend on a priorly chosen form of the mapping. We have demonstrated the feasibility of this approach in two examples, linear and locally linear projections as induced by the t-SNE cost function. Interestingly, it is possible to also integrate auxiliary (e.g. class) information into the framework.

This general view opens the way towards alternatives since, in principle, every cost function can be combined with every possible form of the mapping function. Even more interesting, the framework allows us to consider the generalization ability of dimensionality reduction since an explicit cost function is available in terms of the reconstruction errore. Interestingly, bounds as derived in the context of computational learning theory can directly be transferred to this setting.

The investigation of alternative dimensionality reduction mappings including more global cost functions such as provided by Isomap, and locally non-linear function approximations, as well as the derivative of explicit bounds on its generalization ability will be the subject of future work.

Acknowledgment

This work was supported by the ”Nederlandse organisatie voor Wetenschappelijke Onderzoek (NWO)“ under project code 612.066.620 and by the ”German Science Foundation (DFG)“ under grant number HA-2719/4-1. Further, financial support from the Cluster of Excellence 277 Cognitive Interaction Technology funded in the framework of the German Excellence Initiative is gratefully acknowledged.

References

[1] J. Lee and M. Verleysen, *Nonlinear dimensionality reduction*, 1st ed. Springer, 2007.

[2] L. J. P. van der Maaten, E. O. Postma, and H. J. van den Herik, “Dimensionality reduction: A comparative review,” Tilburg University, Tech. Rep. TiCC-TR 2009-005, Oct 2009.

[3] L. van der Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, November 2008.

[4] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler, “Visual analytics: Scope and challenges,” in *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*, S. Simoff, M. H. Boehlen, and A. Mazeika, Eds. Springer, 2008, lecture Notes in Computer Science (LNCS).

[5] J. Venna, J. Peltonen, K. Nybo, H. Aidos, and S. Kaski, “Information retrieval perspective to nonlinear dimensionality reduction for data visualization,” *J. Mach. Learn. Res.*, vol. 11, pp. 451–490, 2010.

[6] S. T. Roweis and L. K. Saul, “Nonlinear Dimensionality Reduction by Locally Linear Embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[7] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, “A Global Geometric Framework for Nonlinear Dimensionality Reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[8] M. Belkin and P. Niyogi., “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Computation*, vol. 15, pp. 1373–15 396, 2003.

[9] K. Q. Weinberger and L. K. Saul, “An introduction to nonlinear dimensionality reduction by maximum variance unfolding,” *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006.

[10] G. Hinton and S. Roweis, “Stochastic neighbor embedding,” in *Advances in Neural Information Processing Systems 15*. MIT Press, 2003, pp. 833–840.

[11] M. Á. Carreira-perpiñán, “The elastic embedding algorithm for dimensionality reduction,” in *27th Int. Conf. Machine Learning (ICML 2010)*, 2010, pp. 167–174.

[12] L. J. P. van der Maaten, “Learning a parametric embedding by preserving local structure,” in *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AI-STATS)*, no. 5. JMLR W&CP, 2009, pp. 384–391.

[13] W. Torgerson, “Multidimensional scaling, i: Theory and method,” *Psychometrika*, vol. 17, pp. 401–419, 1952.

[14] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2007.

[15] M. Brand, “Charting a manifold,” Mitsubishi Electric Research Laboratories (MERL), Tech. Rep. 15, 2003.

[16] Y. Teh and S. Roweis, “Automatic alignment of local representations,” *Advances in Neural Information Processing Systems*, vol. 15, pp. 841–848, 2003.

[17] C. M. Bishop and C. K. I. Williams, “Gtm: The generative topographic mapping,” *Neural Computation*, vol. 10, pp. 215–234, 1998.

[18] T. Kohonen, *Self-organizing Maps*. Springer, 1995.

[19] Y. W. Teh and S. Roweis, “Automatic alignment of local representations,” in *In Advances in Neural Information Processing Systems 15*. MIT Press, 2003, pp. 841–848.

[20] J. A. K. Suykens, “Data visualization and dimensionality reduction using kernel maps with a reference point,” *Neural Networks, IEEE Transactions on*, vol. 19, no. 9, pp. 1501 –1517, sept. 2008.

[21] J. A. Lee and M. Verleysen, “Rank-based quality assessment of nonlinear dimensionality reduction,” in *16th European Symposium on Artificial Neural Networks (ESANN)*, Bruges, Belgium, 2008, pp. 49–54.

[22] ——, “Quality assessment of dimensionality reduction: Rank-based criteria,” *Neurocomput.*, vol. 72, no. 7-9, pp. 1431–1443, 2009.

[23] R. Möller and H. Hoffmann, “An extension of neural gas to local pca,” *Neurocomputing*, vol. 62, no. 305-326, 2004.

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

[25] M. Cottrell, B. Hammer, A. Hasenfuss, and T. Villmann, “Batch and median neural gas,” *Neural Networks*, vol. 19, pp. 762–771, 2006.

[26] “Limited rank matrix learning: Discriminative dimension reduction and visualization,” submitted to Neural Networks.

[27] P. L. Bartlett and S. Mendelson, “Rademacher and gaussian complexities: risk bounds and structural results,” *J. Mach. Learn. Res.*, vol. 3, pp. 463–482, 2003.