# Parametric nonlinear dimensionality reduction using kernel t-SNE

Andrej Gisbrecht Alexander Schulz
Barbara Hammer
University of Bielefeld - CITEC centre of excellence, Germany

Preprint of the publication [10], as provided by the authors.

**Abstract**

Novel non-parametric dimensionality reduction techniques such as t-distributed stochastic neighbor embedding (t-SNE) lead to a powerful and flexible visualization of high-dimensional data. One drawback of non-parametric techniques is their lack of an explicit out-of-sample extension. In this contribution, we propose an efficient extension of t-SNE to a parametric framework, kernel t-SNE, which preserves the flexibility of basic t-SNE, but enables explicit out-of-sample extensions. We test the ability of kernel t-SNE in comparison to standard t-SNE for benchmark data sets, in particular addressing the generalization ability of the mapping for novel data. In the context of large data sets, this procedure enables us to train a mapping for a fixed size subset only, mapping all data afterwards in linear time. We demonstrate that this technique yields satisfactory results also for large data sets provided missing information due to the small size of the subset is accounted for by auxiliary information such as class labels, which can be integrated into kernel t-SNE based on the Fisher information.

## 1 Introduction

Handling big data constitutes one of the main challenges of information technology in the new century, incorporating, among other issues, the task to create 'effective human-computer interaction tools for facilitating rapidly customizable visual reasoning for diverse missions' [13]. In this context, the visual inspection of high-dimensional data sets offers an intuitive interface for humans to rapidly detect structural elements of the data such as clusters, homogeneous regions, or outliers, relying on the astonishing cognitive capabilities of humans for instantaneous visual perception of structures and grouping of items [34].

Dimensionality reduction (DR) refers to the problem to map high-dimensional data points to low dimensions such that as much structure as possible is preserved. Starting with classical methods such as principal component analysis (PCA), multidimensional scaling (MDS), or the self-organizing map (SOM), it offers a visual data analysis tool which has been successfully used in diverse areas such as social sciences or bioinformatics since decades [15, 28]. In the last years, a huge variety of diverse alternative DR techniques has emerged, including popular algorithms such as locally linear embedding (LLE), Isomap, Isotop, maximum variance unfolding (MVU), Laplacian eigenmaps, neighborhood retrieval visualizer, maximum entropy unfolding, t-distributed stochastic neighbor embedding (t-SNE), and many others [23, 26, 35, 3, 31, 33], see e.g. [32, 33, 17, 6] for overviews. These methods belong to nonlinear DR techniques, enabling the correct visualization of data which lie on curved manifolds or which incorporate clusters of complex shape, as is often the case for real-life examples, thus opening the way towards a visual inspection of nonlinear phenomena in the given data.

Unlike the classical techniques PCA and SOM, most recent DR methods belong to the class of non-parametric techniques: they provide a mapping of the given data points only, without an explicit mapping prescription how to project further points which are not contained in the data set to low dimensions. This choice has the benefit that it equips the techniques with a high degree of flexibility: no constraints have to be met due to a predefined form of the mapping, rather, depending on the situation at hand, arbitrary restructuring, tearing, or nonlinear transformation of data is possible. Hence, these techniques carry the promise to arrive at a very flexible visualization of data such that also subtle nonlinear structures can be spotted. Naturally, this flexibility comes at a price to pay: (i) the result of the visualization step entirely depends on the way in which the mapping procedure is formalized, such that, depending on the chosen technique, very different results can be obtained. Commonly, all techniques necessarily have to take information loss into account when projecting high-dimensional data onto lower dimensions. The way in which a concrete method should be interpreted and which aspects are faithfully visualized, which aspects, on the contrary, are artefacts of the projection is not always easily accessible to applicants due to the diversity of existing techniques. (ii) There does not exist a direct way to map additional data points after having obtained the projection of the given set. This fact makes the technique unsuitable for the visualization of streaming data or online scenarios. Further, it prohibits a visualization of parts of a given data set only, extending to larger sets on demand. The latter strategy, however, would be vital if large data sets are dealt with: all modern nonlinear non-parametric DR techniques display an at least quadratic complexity, which makes them unsuitable for large data sets already in the range of about 10,000 data points with current desktop computers. Efficient approximation techniques with better efficiency are just popping up recently [30, 36]. Thus, it would be desirable, to map a part first, to obtain a rough overview, zooming in the details on demand.

These two drawbacks have the consequence that classical techniques such as PCA or SOM are still often preferred in practical applications: Both, PCA and SOM rely on very intuitive principles as regards both, learning algorithms and their final result. They capture directions in the data of maximum variance, globally for PCA and locally for SOM. Online learning algorithms such as online SOM training or the Oja learning rule mimic fundamental principles as found in the human brain, being based on the Hebbian principle accompanied by topology preservation in case of SOM [15]. In addition to this intuitive training procedure and outcome, both techniques have severe practical benefits: training can be done efficiently in linear time only, which is a crucial prerequisite if large data sets are dealt with. In addition, both techniques do not only project the given data set, but they offer an explicit mapping of the full data space to two dimensions by means of an explicit linear mapping in case of PCA and a winner takes all mapping based on prototypes in case of SOM. Further, for both techniques, online training approaches which are suitable for streaming data or online data processing, exist. Therefore, despite the larger flexibility of many modern non-parametric DR techniques, PCA and SOM still by far outnumber these alternatives regarding applications.

In this contribution, to address this gap,we discuss recent developments connected to the question of how to turn non-parametric dimensionality reduction techniques into parametric approaches without losing the underlying flexibility. In particular, we introduce kernel t-SNE as a flexible approach with a particularly simple training procedure. We demonstrate, that kernel t-SNE maintains the flexibility of t-SNE, and that it displays excellent generalization ability within out-of-sample extensions.

This approach opens the way towards endowing t-SNE with linear complexity: we can train t-SNE on a small subset of fixed size only, mapping all data in linear time afterwards. We will show that the flexibility of the mapping can result in problems in this case: while subsampling, only a small part of the information of the full data set is used. In consequence, the data projection can be sub-optimum due to the missing information to shape the ill-posed problem of dimensionality reduction. Here, an alternative can be taken: we can enhance the information content of the data set without enlarging the computational complexity by taking auxiliary information into account. This way, the visualization can concentrate on the aspects relevant for the given auxiliary information rather than potential noise. In addition, this possibility opens the way towards a better interpretability of the results, since the user can specify the relevant aspects for the visualization in an explicit way. One specific type of auxiliary information which is often available in applications is offered by class labeling.

There exist quite a few approaches to extend DR techniques to incorporate auxiliary class labels: classical linear ones include Fisher's linear discriminant analysis, partial least squares regression, or informed projections, for example [7, 17]. These techniques can be extended to nonlinear methods by means of kernelization [19, 2]. Another principled way to extend dimensionality reducing data visualization to auxiliary information is offered by an adaptation of the underlying metric. The principle of learning metrics has been introduced in [14, 21]: the standard Riemannian metric is substituted by a form which measures the information of the data for the given classification task. The Fisher information matrix induces the local structure of this metric and it can be expanded globally in terms of path integrals. This metric is integrated into SOM, MDS, and a recent information theoretic model for data visualization [14, 21, 33]. A drawback of the proposed method is its high computational complexity. Here, we circumvent this problem by integrating the Fisher metric for a small training set only, enabling the projection of the full data set by means of an explicit nonlinear mapping. This way, very promising results can be obtained also for large data sets.

Now, we will first shortly review popular dimensionality reduction techniques, in particular t-SNE in more detail. Afterwards, we address the question how to enhance non-parametric techniques towards an explicit mapping prescription, emphasizing kernel t-SNE as one particularly flexible approach in this context. Finally, we consider discriminative dimensionality reduction based on the Fisher information, testing this principle in the context of kernel t-SNE.

## 2 Dimensionality reduction

Assume a high-dimensional input space $X$ is given, e.g. $X \subset \mathbb{R}^N$ constitutes a data manifold for which a sample of points is available. Data

$\mathbf{x}_i, i = 1, \ldots, m$ in $X$ should be projected to points $\mathbf{y}_i, i = 1, \ldots, m$ in the projection space $Y = \mathbb{R}^2$ such that as much structure as possible is preserved. The notion of 'structure preservation' is ill-posed and many different mathematical specifications of this term have been used in the literature. One of the most classical algorithms is PCA which maps data linearly to the directions with largest variance, corresponding to the eigenvectors with largest eigenvalues of the data covariance matrix.

PCA constitutes one of the most fundamental approaches and one example of two different underlying principles [27]: (i) PCA constitutes the linear transformation which allows the best reconstruction of the data from its low dimensional projection in a least squares sense. That means, assuming centered data, it optimizes the objective $\sum_i (\mathbf{x}_i - W(W^t\mathbf{x}_i))^2$ with respect to the parameters of the low-dimensional linear mapping $\mathbf{x} \to \mathbf{y} = W^t\mathbf{x}$. (ii) PCA tries to find the linear projections of the points such that the variance in these directions is maximized. Alternatively speaking, since the variance of the projections is always limited by the variance in the original space, it tries to preserve as much variance of the original data set as compared to its projection as possible. The first motivation treats PCA as a generative model, the latter as a cost minimizer. Due to the simplicity of the underlying mapping, the results coincide.

This is, however, not the case for general nonlinear approaches. Roughly speaking, there exist two opposite ways to introduce dimensionality reduction, which together cover most existing DR approaches: (i) the generative, often parametric approach, which takes the point of view that high-dimensional data points are generated by or reconstructed from a low-dimensional structure which can be visualized directly, (ii) and the cost-function based, often non-parametric approach, which, on the opposite, tries to find low-dimensional projection points such that the characteristics of the original high-dimensional data are preserved as much as possible. Popular models such as PCA, SOM, its probabilistic counterparts the probabilistic PCA or the generative topographic mapping (GTM), and encoder frameworks such as deep autoencoder networks fall under the first, generative framework [17, 32, 4]. The second framework can cover diverse modern non-parametric approaches such as Isomap, MVU, LLE, SNE, or t-SNE, as recently demonstrated in the overview [6].

### A note on parametric approaches

Parametric approaches are often less flexible as compared to non-parametric ones since they rely on a fixed priorly specified form of the DR mapping.

Depending on the form of the parametric mapping, constraints have to be met. This is particularly pronounced for linear mappings, but also nonlinear generalizations such as SOM or GTM heavily depend on inherent constraints induced by the prototype-based modeling of the data. Note that a few alternative manifold learners have been proposed, partially on top of non-parametric approaches, which try to find an explicit model of the data manifold and usually provide a projection mapping of the data into low dimensions: examples include tangent space intrinsic manifold regularization [25], manifold charting [5] or corresponding extensions of powerful prototype based techniques such as matrix learning neural gas [1]. Manifold coordination also takes place in parametric extensions of non-parametric approaches such as proposed in locally linear coordination [22]. However, these techniques rely on an intrinsically low-dimensional manifold and they are less suited to extend modern nonlinear projection techniques which can also cope with information loss.

Note that not only an explicit mapping, but usually also an approximate inverse is given for such methods: for PCA, it is offered by the transposed of the matrix; for SOM and GTM, it is given by the explicit prototypes or centres of the Gaussians which are points in the data space; for auto-encoder networks, an explicit inverse mapping is trained simultaneously to the embedding; generalizations of PCA towards local techniques allow at least a local inverse of the mapping [1]. Due to this fact, a very clear objective of the techniques can be formulated in the form of the data reconstruction error. Based on this observation, a training technique which minimizes this reconstruction error or a related quantity can be derived. This fact often makes the methods and their training intuitively interpretable. Besides this fact, an explicit mapping prescription allows direct out-of-sample extensions, online, and life-long training of the mapping prescription.

In particular for streaming data, very large data sets, or online scenarios, this fact allows the user to adapt the mapping on only a part of the data set and to display a part of the data on demand, thereby controlling the efficiency and stationarity of the resulting mapping by means of the amount of data taken into account.

Albeit classical parametric methods have been developed for vectorial data only, a variety of extensions has been proposed in the last years, which rely on pairwise distances of data rather than an explicit vectorial representation. Examples include, in particular, kernel and relational variants of SOM and GTM [37, 11, 12]. Due to their dependence on a full distance matrix, these techniques have inherent quadratic complexity if applied for the full data set. Here, an explicit mapping and a corresponding strategy to iteratively train the mapping on parts of the data only has beneficial effects, since it reduces the complexity to linear one. Thereby, different strategies have been proposed in the literature, in particular patch processing has been proposed which iteratively takes into account all data in terms of compressed prototypes [11, 12].

## Nonparametric approaches

Nonparametric methods often take a simple cost function based approach: the data points $\mathbf{x}_i$ contained in a high-dimensional vector space constitute the starting point; for every point coefficients $\mathbf{y}_i$ are determined in $Y$ such that the characteristics of these points mimic the characteristics of their high-dimensional counterpart. Thereby, the characteristics differ from one method to the other, referring e.g. to pairwise distances of data, the data variation, locally linear relations of data points, or local probabilities induced by the pairwise distances, to name a few examples.

We consider t-SNE [31] in more detail, since it demonstrates the strengths and weaknesses of this principle in an exemplary way. Probabilities in the original space are defined as $p_{ij} = (p_{(i|j)} + p_{(j|i)})/(2m)$ where

$$p_{j|i} = \frac{\exp(-0.5\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma_i^2)}{\sum_{k,k\neq i}\exp(-0.5\|\mathbf{x}_i - \mathbf{x}_k\|^2/\sigma_i^2)}$$

depends on the pairwise distances of points; $\sigma_i$ is automatically determined by the method such that the effective number of neighbors coincides with a priorly specified parameter, the perplexity. In the projection space, probabilities are induced by the Student t-distribution

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_k \sum_{l,l\neq k}(1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}$$

to avoid the crowding problem by using a long tail distribution. The goal is to find projections $\mathbf{y}_i$ such that the difference between $p_{ij}$ and $q_{ij}$ becomes small as measured by the Kullback-Leibler divergence. t-SNE relies on a gradient based technique.

Many alternative non-parametric techniques proposed in the literature have a very similar structure, as pointed out in [6]: They extract a characteristic of the data points $\mathbf{x}_i$ and try to find projections $\mathbf{y}_i$ such that the corresponding characteristics are as close as possible as measured by some cost function. [6] summarizes some of today's most popular dimensionality reduction methods this way. In the following, we will exemplarily consider the alternatives maximum variance unfolding (MVU), locally linear embedding (LLE), and Isomap. The ratio behind these methods is the following: MVU aims at a maximization of the variance of the projected points such that the distances are preserved for local neighborhoods of every point. This problem can be formalized by means of a quadratic optimization problem [35]. LLE represents points in terms of linear combinations of its local neighborhood and tries to find projections such that these relations remain valid. Thereby, problems are formalized as a quadratic optimization task such that an explicit algebraic solution in terms of eigenvalues is possible [23]. Isomap constitutes an extension of classical multidimensional scaling which approximates the manifold distances in the data space by means of geodesic distances. After having done so, the standard eigenvalue decomposition of the corresponding similarities allows an approximate projection to two dimensions [26].

These techniques do not rely on a parametric form such that they display a rich flexibility to emphasize local nonlinear structures. This makes them much more flexible as compared to linear approaches such as PCA, and it can also give fundamentally different results as compared to GTM or SOM, which are constrained to inherently smooth mappings. This flexibility is payed for by two drawbacks, which make the techniques unsuited for large data sets: (i) The techniques do not provide direct out-of-sample extensions, (ii) the techniques display at least quadratic complexity. Thus, these methods are not suited for large data sets in their direct form.

## 3 Kernel t-SNE

How to extend a non-parametric dimensionality reduction technique such as t-SNE to an explicit mapping? We fix a parametric form $\mathbf{x} \to f_w(\mathbf{x}) = \mathbf{y}$ and optimize the parameters of $f_w$ instead of the projection coordinates. Such an extension of non-parametric approaches to a parametric version has been proposed in [29, 6, 9] in different forms. In [29], $f_w$ takes the form of deep-autoencoder networks, which are trained in two steps: first, the deep auto-encoder is trained in a standard way to encode the given examples; afterwards, parameters are fine tuned such that the t-SNE cost function is optimized when plugging the images of given data points into the mapping. Due to the high flexibility of deep networks, this method achieves good results provided enough data are present and training is done in an accurate way. Due to the large number of parameters of deep auto-encoders, the resulting mapping is usually of very complex form, and its training requires a large number of data and large training time. In [6] the principle of plugging a parametric form $f_w$ in any cost function based non-parametric DR techniques is elucidated, and it is tested in the context of t-SNE with linear or piecewise linear functions. Due to the simplicity of these functions, a very good generalization is obtained already on small data sets, and the training time is low. However, the flexibility of the resulting mapping is restricted as compared to full t-SNE since local nonlinear phenomena cannot be captured by locally linear mappings. In [9], already first steps into the direction of kernel t-SNE have been proposed: the mapping $f_w$ is given by a linear combination of Gaussians, where the coefficients are trained based on the t-SNE cost function, or in a direct way by means of the pseudo-inverse of a given training set, mapped using t-SNE. Surprisingly, albeit being much simpler, the latter technique yields comparable results, as investigated in [9]. We will see that this latter training technique also opens the way towards an efficient integration of auxiliary information by means of Fisher kernel t-SNE. Due to this fact, we follow the approach in [9] and use a normalized form of such a kernel mapping together with a particularly efficient direct training technique.

The mapping $f_w = \mathbf{y}$ underlying kernel t-SNE has the following form:

$$\mathbf{x} \mapsto \mathbf{y}(\mathbf{x}) = \sum_j \boldsymbol{\alpha}_j \cdot \frac{k(\mathbf{x}, \mathbf{x}_j)}{\sum_l k(\mathbf{x}, \mathbf{x}_l)}$$

where $\boldsymbol{\alpha}_j \in Y$ are parameters corresponding to points in the projection space and the data $\mathbf{x}_j$ are taken as a fixed sample, usually $j$ runs over a small subset $X'$ sampled from the data $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$. $k$ is the Gaussian kernel parameterized by the bandwidth $\sigma_j$:

$$k(\mathbf{x}, \mathbf{x}_j) = \exp(-0.5\|\mathbf{x} - \mathbf{x}_j\|^2/\sigma_j^2)$$

In the limit of small bandwidth, original t-SNE is resembled for inputs taken from the points $X'$ of the sum. For these points, in the limit, the parameter $\boldsymbol{\alpha}_j$ corresponds to the projected $\mathbf{y}_j$ of $\mathbf{x}_j$. For other points $\mathbf{x}$, an interpolation takes place according to the relative distance of $\mathbf{x}$ from samples $\mathbf{x}_i$ in $X'$.

Note that this mapping constitutes a generalized linear mapping such that training can be done in a particularly simple way provided a set of samples $\mathbf{x}_i$ and $\mathbf{y}(\mathbf{x}_i)$ is available. Then the parameters $\boldsymbol{\alpha}_j$ can be analytically determined as the least squares solution of the mapping: Assume $\mathbf{A}$ contains the parameter vectors $\boldsymbol{\alpha}_j$ in its rows, $\mathbf{K}$ is the normalized Gram matrix with entries

$$[\mathbf{K}]_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j) / \sum_l k(\mathbf{x}_i, \mathbf{x}_l)$$

and $\mathbf{Y}$ denotes the matrix of projections $\mathbf{y}_i$ (also as its rows). Then, a minimum of the least squares error

$$\sum_i \|\mathbf{y}_i - \mathbf{y}(\mathbf{x}_i)\|^2$$

with respect to the parameters $\boldsymbol{\alpha}_j$ has the form

$$\mathbf{A} = \mathbf{K}^{-1} \cdot \mathbf{Y}$$

where $\mathbf{K}^{-1}$ refers to the pseudo-inverse of $\mathbf{K}$.

For kernel t-SNE, we use standard t-SNE for the subset $X'$ to obtain a training set. Afterwards, we use this explicit analytical solution to obtain the parameters of the mapping. Having obtained the mapping, the full set $X$ can be projected in linear time by applying the mapping $\mathbf{y}$. Obviously, it is possible to extend alternative dimensionality reduction techniques such as Isomap, LLE, or MVU directly in the same way. We refer to the resulting mapping in terms of kernel Isomap, kernel LLE, and kernel MVU, respectively.

The bandwidth $\sigma_i$ of the mapping constitutes a critical parameter of the mapping since it determines the smoothness and flexibility of the resulting kernel mapping. We use a principled approach to determine this parameter as follows: $\sigma_i$ is chosen as a multiple of the distance of $\mathbf{x}_i$ from its closest neighbor in $X'$, where the scaling factor is typically taken as a small positive value. We determine this factor automatically as the smallest value in such a way that all entries of $\mathbf{K}$ are within the range of representable numbers (resp. a predefined interval).

Algorithm 1 summarizes the kernel t-SNE method. The matrix $\mathbf{X}$ contains all the data vectors in its rows. The method SELECTTRAININGSET randomly selects a subset of the data of size $nTrain$ for the training of the mapping. In section 6 we investigate which size is a proper choice. The method CALCPAIRWISEDIS calculates pairwise distances between all points in the given data matrices. TSNE performs the t-SNE algorithm on the training set with the perplexity parameter *perpl*. Finally, the method DETERMINESIGMA selects the $\sigma_i$ parameters for the kernels as described previously.

## 4 Discriminative dimensionality reduction

Kernel t-SNE enables to map large data sets in linear time by training a mapping on a small subsample only, yielding acceptable results. However, it

**Algorithm 1** kernel t-SNE

```
function $\textsc{ktsne}(\mathbf{X}, nTrain, perpl)$
    $(\mathbf{X}_{tr}, \mathbf{X}_{test}) = \textsc{selectTrainingSet}(\mathbf{X}, nTrain)$
    $\mathbf{D}_{tr} = \textsc{calcPairwiseDis}(\mathbf{X}_{tr}, \mathbf{X}_{tr})$
    $\mathbf{D}_{test} = \textsc{calcPairwiseDis}(\mathbf{X}_{tr}, \mathbf{X}_{test})$
    $\mathbf{Y}_{tr} = \textsc{tsne}(\mathbf{D}_{tr}, perpl)$
    $\boldsymbol{\sigma} = \textsc{determineSigma}(\mathbf{D}_{tr})$
    for all entries $(i, j)$ from $\mathbf{D}_{tr}$ do
        $[\mathbf{K}]_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j) / \sum_l k(\mathbf{x}_i, \mathbf{x}_l)$
    end for
    $\mathbf{A} = \mathbf{K}^{-1} \cdot \mathbf{Y}_{tr}$
    for all entries $(i, j)$ from $\mathbf{D}_{test}$ do
        $[\mathbf{K}]_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j) / \sum_l k(\mathbf{x}_i, \mathbf{x}_l)$
    end for
    $\mathbf{Y}_{test} = \mathbf{K} \cdot \mathbf{A}$
    return $(\mathbf{Y}_{tr}, \mathbf{Y}_{test})$
end function
```

is often the case that the underlying data structure such as cluster formation is not yet as pronounced based on a small subset only as it would be for the full data set. Thus, albeit kernel t-SNE shows excellent generalization ability, the results are different as compared to t-SNE when applied for the full data set due to missing information in the data used for training of the map. How can this information gap be closed?

It has been proposed in [14, 21, 33] to enrich nonlinear dimensionality reduction techniques such as the self-organizing map by auxiliary information in order to enforce the method to display the information which is believed as relevant by an applicant. A particularly intuitive situation is present if data are enriched by accompanying class labels, and the information most relevant for the given classification at hand should be displayed. We follow this approach and devise a particularly simple method to incorporate this information into the mapping based on kernel t-SNE.

Formally, we assume that every data point $\mathbf{x}_i$ is equipped with a class label $c_i$. Projection points $\mathbf{y}_i$ should be found such that the aspects of $\mathbf{x}_i$ which are relevant for $c_i$ are displayed.

From a mathematical point of view, this auxiliary information can be easily integrated into a projection technique by referring to the Fisher information, as detailed e.g. in [21]. We consider the Riemannian manifold spanned by the data points $\mathbf{x}_i$. Each point $\mathbf{x}$ is equipped with a local Riemannian tensor $\mathbf{J}(\mathbf{x})$ which is used to define a scalar product $g_{\mathbf{x}}$ between two tangent vectors $\mathbf{u}$ and $\mathbf{v}$ on the manifold at position $\mathbf{x}$:

$$g_{\mathbf{x}}(\mathbf{u},\mathbf{v}) = \mathbf{u}^T \mathbf{J}(\mathbf{x})\mathbf{v}.$$

The local Fisher information matrix $\mathbf{J}(\mathbf{x})$ is computed via

$$\mathbf{J}(\mathbf{x}) = E_{p(c|\mathbf{x})}\left\{ \left(\frac{\partial}{\partial \mathbf{x}} \log p(c|\mathbf{x})\right) \left(\frac{\partial}{\partial \mathbf{x}} \log p(c|\mathbf{x})\right)^T \right\}.$$

Thereby, $E$ denotes the expectation, and $p(c|\mathbf{x})$ refers to the probability of class $c$ given the data point $\mathbf{x}$. Essentially, this tensor locally scales dimensions in the tangent space in such a way that exactly those dimensions are amplified which are relevant for the given class information.

A Riemannian metric is induced by this local quadratic form in the classical way, we refer to this metric as the Fisher metric in the following: For given points $\mathbf{x}$ and $\mathbf{x}'$ on the manifold, the distance is

$$d(\mathbf{x},\mathbf{x}') = \inf_{\gamma} \int_0^1 \sqrt{g_{\gamma(t)}(\gamma'(t),\gamma'(t))}dt$$

where $\gamma : [0,1] \to X$ ranges over all smooth paths with $\gamma(0) = \mathbf{x}$ to $\gamma(1) = \mathbf{x}'$ in $X$. We refer to this metric as the Fisher metric in the following. This metric measures distances between data points $\mathbf{x}$ and $\mathbf{x}'$ along the Riemannian manifold, thereby locally transforming the space according to its relevance for the given label information. It can be shown that this learning metrics principle refers to the information content of the data with respect to the given auxiliary information as measured locally be the Kullback-Leibler divergence [14].

There are two problems to this approach: first, how to compute this learning metrics efficiently for a given labeled data set? In practice, the probability $p(c|\mathbf{x})$ is not known. Further, optimum path integrals cannot be efficiently computed analytically. Second, how can we efficiently integrate this learning metrics principle into kernel t-SNE?

## Efficient computation of the Fisher metric

In practice, the Fisher distance has to be estimated based on the given data only. The conditional probabilities $p(c|\mathbf{x})$ can be estimated from the data using the Parzen nonparametric estimator

$$\hat{p}(c|\mathbf{x}) = \frac{\sum_i \delta_{c=c_i} \exp(-0.5\|\mathbf{x}-\mathbf{x}_i\|^2/\sigma^2)}{\sum_j \exp(-0.5\|\mathbf{x}-\mathbf{x}_j\|^2/\sigma^2)}.$$

The Fisher information matrix becomes

$$\mathbf{J}(\mathbf{x}) = \frac{1}{\sigma^4} E_{\hat{p}(c|\mathbf{x})} \left\{ \mathbf{b}(\mathbf{x}, c)\mathbf{b}(\mathbf{x}, c)^T \right\}$$

where

$$\mathbf{b}(\mathbf{x}, c) = E_{\xi(i|\mathbf{x},c)}\{\mathbf{x}_i\} - E_{\xi(i|\mathbf{x})}\{\mathbf{x}_i\}$$

$$\xi(i|\mathbf{x}, c) = \frac{\delta_{c,c_i} \exp(-0.5\|\mathbf{x} - \mathbf{x}_i\|^2/\sigma^2)}{\sum_j \delta_{c,c_j} \exp(-0.5\|\mathbf{x} - \mathbf{x}_j\|^2/\sigma^2)}$$

$$\xi(i|\mathbf{x}) = \frac{\exp(-0.5\|\mathbf{x} - \mathbf{x}_i\|^2/\sigma^2)}{\sum_j \exp(-0.5\|\mathbf{x} - \mathbf{x}_j\|^2/\sigma^2)}$$

$E$ denotes the empirical expectation, i.e. weighted sums with weights depicted in the subscripts. If large data sets or out-of-sample extensions are dealt with, a subset of the data only is usually sufficient for the estimation of $\mathbf{J}(\mathbf{x})$.

There exist different ways to approximate the path integrals based on the Fisher matrix as discussed in [21]. An efficient way which preserves locally relevant information is offered by $T$-approximations: $T$ equidistant points on the line from $\mathbf{x}_i$ to $\mathbf{x}_j$ are sampled, and the Riemannian distance on the manifold is approximated by

$$d_T(\mathbf{x}_i, \mathbf{x}_j) = \sum_{t=1}^{T} d_1 \left( \mathbf{x}_i + \frac{t-1}{T}(\mathbf{x}_j - \mathbf{x}_i), \mathbf{x}_i + \frac{t}{T}(\mathbf{x}_j - \mathbf{x}_i) \right)$$

where $d_1(\mathbf{x}_i, \mathbf{x}_j) = g_{\mathbf{x}_i}(\mathbf{x}_i - \mathbf{x}_j, \mathbf{x}_i - \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T J(\mathbf{x}_i)(\mathbf{x}_i - \mathbf{x}_j)$ is the standard distance as evaluated in the tangent space of $\mathbf{x}_i$. Locally, this approximation gives good results such that a faithful dimensionality reduction of data can be based thereon.

## Efficient integration of the Fisher metric into kernel t-SNE

In [8], it has been proposed to integrate this Fisher information into kernel t-SNE by means of a corresponding kernel. Here, we take an even simpler perspective: we consider a set of data points $\mathbf{x}_i$ equipped with the pairwise Fisher metric which is estimated based on their class labels taking simple linear approximations for the path integrals. Using t-SNE, a training set $X'$ is obtained which takes the auxiliary label information into account, since pairwise distances of data are computed based on the Fisher metric in this set. We infer a kernel t-SNE mapping as before, which is adapted to the

**Algorithm 2** Fisher kernel t-SNE

```
function fktsne($\mathbf{X}, nTrain, perpl$)
    $(\mathbf{X}_{tr}, \mathbf{X}_{test})$ = selectTrainingSet($\mathbf{X}, nTrain$)
    $\mathbf{D}_{trDisc}$ = calcPairwiseFisherDis($\mathbf{X}_{tr}, \mathbf{X}_{tr}$)
    $\mathbf{D}_{tr}$ = calcPairwiseDis($\mathbf{X}_{tr}, \mathbf{X}_{tr}$)
    $\mathbf{D}_{test}$ = calcPairwiseDis($\mathbf{X}_{tr}, \mathbf{X}_{test}$)
    $\mathbf{Y}_{tr}$ = tsne($\mathbf{D}_{trDisc}, perpl$)
    $\boldsymbol{\sigma}$ = determineSigma($\mathbf{D}_{tr}$)
    for all entries $(i, j)$ from $\mathbf{D}_{tr}$ do
        $[\mathbf{K}]_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j) / \sum_l k(\mathbf{x}_i, \mathbf{x}_l)$
    end for
    $\mathbf{A} = \mathbf{K}^{-1} \cdot \mathbf{Y}_{tr}$
    for all entries $(i, j)$ from $\mathbf{D}_{test}$ do
        $[\mathbf{K}]_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j) / \sum_l k(\mathbf{x}_i, \mathbf{x}_l)$
    end for
    $\mathbf{Y}_{test} = \mathbf{K} \cdot \mathbf{A}$
    return $(\mathbf{Y}_{tr}, \mathbf{Y}_{test})$
end function
```

label information due to the information inherent in the training set. The resulting map is adapted to the relevant information since this information is encoded in the training set. We refer to this technique as Fisher kernel t-SNE in the following.

Algorithm 2 details the resulting procedure. Again, calcPairwiseDis calculates the pairwise Euclidean distance between all points in the given matrices. calcPairwiseFisherDis calculates the Fisher distance given by $d_T(\mathbf{x}_i, \mathbf{x}_j)$ for each pair. The major difference to kernel t-SNE is that the t-SNE projection is based upon the Fisher distances, while the kernel values in $\mathbf{K}$ are still computed based on the Euclidean metric. As a consequence, Fisher distances do not need to be computed for projections of new points yielding fast out of sample extensions.

## 5 Evaluation measures

Dimensionality reduction being ill-posed, it eventually depends on the task at hand which results are considered as optimum. Nevertheless, formal quantitative measures are vital to enable a comparison of different techniques and an optimization of model meta-parameters based on this general objective.

In the last years, there has been great effort in developing such a baseline, culminating in the formal co-ranking framework as proposed by Lee and Verleysen, which summarizes a variety of different earlier approaches under one common hat [16]. Albeit there are intuitive possibilities to extend this proposal [20], we will stick to this measure in this contribution.

Here, we do not introduce the full co-ranking matrix as given in [16], rather we restrict to the resulting quantitative value referred to as quality in [16]. Essentially, it is generally accepted that a dimensionality reduction technique should preserve neighborhoods of data points in the sense that close points stay close and far away points stay apart. Thereby, the precise distances are less important as compared to the relative ranks. In addition, the exact size of the neighborhood one is interested in depends very much on the situation at hand, usually some small to medium sized range is in the focus of interest. Because of these considerations, it is proposed in [16] to determine the $k$ nearest neighbors for every point $\mathbf{x}_i$ in the original space and the $k$ nearest neighbors of the corresponding projections $\mathbf{y}_i$ in the projection space. Now it is counted, how many indices coincide in these two sets, i.e. how many neighbors stay the same. This is normalized by the baseline $km$, $m$ being the number of points, and averaged over all data points. A quality value $Q_m(k)$ results.

This procedure yields a curve for every visualization which judges in how far neighborhoods are preserved for a neighborhood size $k$ one is interested in. A value close to 1 refers to a good preservation, the baseline for a random mapping being $k/(m-1)$. However, this evaluation measure has a severe drawback: it is not suited for large data sets, it's computation being $\mathcal{O}(m^2 \log m)$, $m$ being the number of points. For this reason, it is worthwhile to use approximation techniques also for the evaluation of such mappings. A simple procedure can be based on sampling. Instead of the full data set, a small subset of size $M$ is taken and the quality is estimated based on this subset. Then the relation $Q_m(k) \approx Q_M(mk/M)$ holds. Naturally, this procedure has a large variance such that taking the mean over several repetitions is advisable.

Based on the co-ranking matrix, this quality measure produces a curve with qualities for each value of the neighborhood parameter $k$, providing a detailed assessment of quality. However, a single scalar value is often more useful when a comparison of many projections is necessary. For this purpose, the evaluation measure $Q_{local}$ has been proposed in [18] which is based on $Q_m(k)$: $Q_{local}$ averages the quality values for small values of $k$. The interval for this is determined automatically. See [18] for further details.

If auxiliary information such as class labels is available, it is possible to additionally evaluate whether the classes are respected in low dimensions by taking the simple k-nearest neighbor classification error in the projections.

## 6 Experiments

In this section we conduct several experimental investigations in order to better understand the effects of applying the proposed kernel mapping.

- We apply the kernel mapping to four different dimensionality reduction techniques and evaluate the quality. The results indicate that t-SNE achieves superior performance and, therefore, we focus our following experiments to kernel t-SNE.
- We empirically analyze the trade off between size of the training set, required time to compute the projection and the resulting generalization performance of the mapping.
- We analyze the distribution of the projected points: How well does the distribution of the projected training set match the distribution of the out-of-sample set?
- We experimentally evaluate the generalization ability of kernel t-SNE towards novel data and compare it to a current state of the art approach for this purpose: parametric t-SNE [29]. This method has been briefly described in section 3.
- We examine the effect of including Fisher information into the framework, i.e. of Fisher kernel t-SNE.

For the experiments, we utilize the following four data sets.

- The *letter* recognition data set describes distorted images of letters in 20 different fonts. It employs 16 features which are basically statistical measures and edge counts. The data set contains 26 classes, i.e one for each capital letter of the English alphabet. 20,000 data points are available.
- The *mnist* data set contains 60,000 images of handwritten digits, where each image consists of 28 × 28 pixels.
- The *norb* data set contains 48,600 images of toys of five different classes. These images were taken from different perspectives and under six different lighting conditions. The number of pixels of the images is 96 × 96.

- The *usps* data set describes handwritten digits from 0 to 9. Each of these 10 classes consists of 1,100 instances resulting in an overall set of 11,000 points. The digits are encoded in 16 × 16 gray scale images.

### 6.1 Applying the proposed kernel mapping to various non-parametric dimensionality reduction techniques

The proposed kernel mapping is a general concept for out-of-sample extension and hence applicable to many nonlinear dimensionality reduction techniques. We enhance Isomap, LLE, MVU and t-SNE with this kernel mapping and we evaluate the generalization performance exemplary on the usps data set. We use 1,000 data points to train each dimensionality reduction technique and employ our kernel mapping in order to project the remaining 10,000 data points. In Figure 1 the evaluation based on the quality value $Q_m(k)$ is depicted where each projection - the direct projection of the training data as well as the out-of-sample extensions (referred to as 'test' here) - is evaluated and plotted into one figure. In order to be independent of the individual sample sizes and to save computational time, the previously in section 5 described sub-sampling strategy for quality evaluation is used here with 100 points in each repetition.

The first important observation is that the train and the corresponding test curve lie close together. This already gives a first indication of the out-of-sample quality of the proposed method. Globally, t-SNE, Isomap and MVU show a similar quality, while locally t-SNE outperforms the remaining approaches if considering small neighborhood sizes.

### 6.2 Properties of the kernel mapping exemplarily evaluated on kernel t-SNE

In order to systematically investigate the influence of the size of the training set on the projection quality, we evaluate different ratios of the training and test set. For this purpose, we apply kernel t-SNE to the usps data set (since it is the smallest it is possible to project the whole data set). The ratios 1%, 10%, 20%, 30%, ..., 90% are used for the training set and the evaluation of each projection is based on the training set (referred to $Q_{train}$) and its corresponding out-of-sample extension ($Q_{test}$). We employ the scalar evaluation measure $Q_{local}$ since it allows us to compare the qualities of many projections in a single plot. Further, we calculate 10 projections for each training set and average the resulting quality values. The quality is visualized on the left axis of Figure 2. In addition, we depict the required running

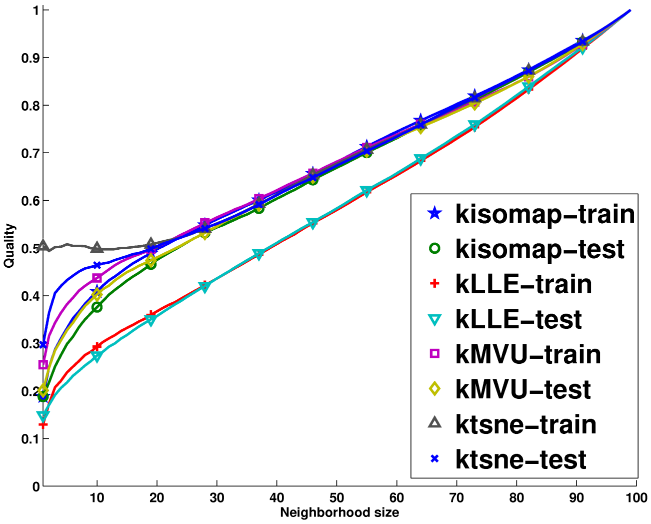

Figure 1: Evaluation of various nonlinear dimensionality reduction approaches together with our proposed kernel mapping on the usps data set.

time on the right coordinate axis.

The quality of the projected training set decreases with increasing training set. This is plausible since the evaluation measure quantifies how well the ranks are preserved and it is obviously easier to preserve ranks if only few data points are available. In this case of very few points, however, the generalization performance degenerates. The quality of the out-of-sample projections stays approximately constant after 10% to 20% while the required computational time grows to the power of two. Consequently, using only 10% of the data for the training set (1100 data points) is enough to obtain a good generalization for the usps data set, as measured by $Q_{local}$.

An interesting question concerning the kernel mapping is the following: How well does the distribution of the projected training set fit the distribution of the out-of-sample extension projected by the kernel mapping? In order to answer this question, we visualize the distribution of the probability values $q_{ij}$ calculated by the t-SNE mapping for the training and test set. For this illustration, we have again used the usps data set. After scaling of both axes (this is necessary due to the different numbers of data points in both data sets), plotting the distribution of the training set above zero and

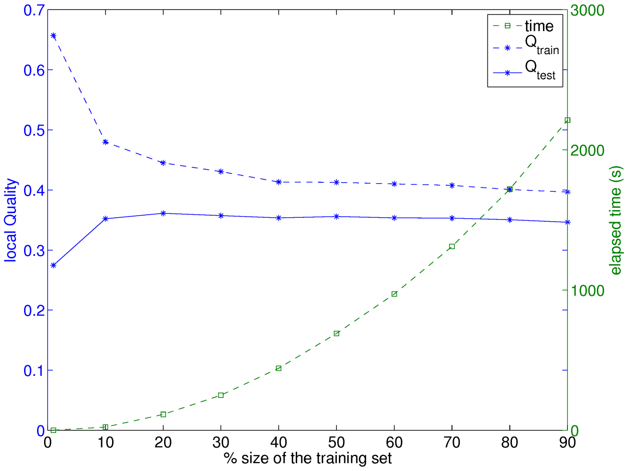

Figure 2: Local qualities $Q_{local}$ and required computational time of the projections based on a varying size of the training set.

the distribution of the test set below (after flipping horizontally) gives the illustration shown in Figure 3. The left image is the original distribution and the right one is zoomed in on the y-axis.

In the left figure we can see that the most probability values are zero. From the right we can deduce statements concerning the similarity of both distributions: the number of probability values in each region is very similar for all regions except the last one. The highest probability value $q_{ij}$ occurs in the test set much more often than in the training set. $q_{ij}$ can be interpreted as the probability that two projected data points $\mathbf{y}_i$ and $\mathbf{y}_j$ are close together. This implies that there are points in the out-of-sample projection which are very close together or lie on top of each other. And indeed, we have observed that some points are projected to the origin. We believe that this is caused by some high-dimensional points lying far apart from all the points of the training set. Managing this issue will be subject to future research.

### 6.3 Comparisons of kernel t-SNE and Fisher kernel t-SNE to parametric t-SNE

Furthermore, we compare the performance of kernel t-SNE to that of parametric t-SNE: we apply both methods on a part of the complete data sets. For usps we utilize 1,000 and for the remaining three data sets 2,000 data

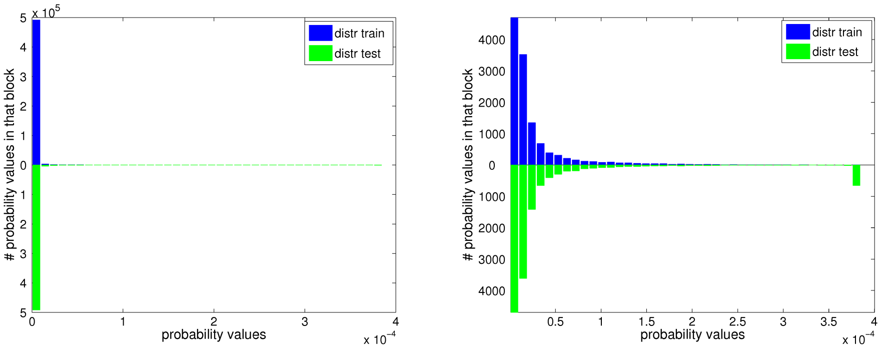

Figure 3: Distribution of the probability values $q_{ij}$ as observed in the training set of t-SNE (above zero) and in the out-of-sample extension (below zero after flipping horizontally). The right figure is zoomed in on the y-axis.

points. Before applying kernel t-SNE, we preprocess the data by projecting them down to 30 dimensions with PCA (for all data sets except letter which is already 16 dimensional). Proceeding similarly as in [29], we do not apply this preprocessing step for parametric t-SNE since the deep architecture of the network used for this method realizes already a preprocessing step by itself. For the application of kernel t-SNE we first train t-SNE on the training set to obtain for each $\mathbf{x}_i$ a two-dimensional point $\mathbf{y}_i$ and then use these pairs to optimize the parameters of our mapping $f_w$ as described in section 3.

Figures 4 and 5 show the resulting projections by kernel t-SNE and parametric t-SNE, respectively. In both cases, the left columns show the projections of the training sets and the right columns those of the complete sets.

We have measured the running time of the two methods on these data sets. This time includes the preprocessing as well as the training and prediction time. Table 1 shows the length of the measured intervals. Kernel t-SNE is usually much faster than parametric t-SNE. This fact can be addressed to the higher training complexity of parametric t-SNE as opposed to kernel t-SNE: while kernel t-SNE relies on an explicit algebraic expression, parametric t-SNE requires the optimization of a cost function induced by t-SNE on the deep autoencoder. For the latter, well-known problems of a classical gradient technique for deep networks prohibit a direct gradient method and pre-training e.g. based on Boltzmann machines is necessary [24].

Table 1: Processing time of kernel t-SNE and parametric t-SNE for all four data sets (in seconds).

| data sets | kernel t-SNE | parametric t-SNE |
|---|---|---|
| letter | 124 | 275 |
| mnist | 145 | 340 |
| norb | 141 | 161 |
| usps | 38 | 126 |

Further, we apply Fisher kernel t-SNE to obtain visualizations which take the labeling of the data into account. Here we also preprocess the data by projecting them to 30 dimensions. The results are depicted in Figure 6.

In order to evaluate the mappings we use the rank based evaluation measure $Q_m(k)$ for different neighborhood sizes $k$ as described in section 5. We use the approximation described in this section, as well: the sample size is fixed to 100 and the evaluation is performed and averaged over ten times. Usually, small to medium values for $k$ are relevant, since they characterize the quality of the local structure preservation.

Figure 7 shows the quality curves for the letter (left) and mnist (right) data sets. For the letter data set, kernel t-SNE shows clearly better results locally than parametric t-SNE, i.e for values of $k$ up to 10 for out-of-sample extension and up to 15 for the training set. For larger values of $k$, parametric t-SNE shows higher accuracy values but as already mentioned before, smaller values of $k$ are usually more important since they characterize the quality of the local structure preservation. Concerning the generalization of kernel t-SNE, the quality curve of the out-of-sample extension lies slightly lower than the one of the training set but approaches the latter with increasing neighborhood range. The training and test curves of Fisher kernel t-SNE proceed similarly as those of kernel t-SNE but lie a bit lower.

The quality curves for the mnist data set are all very close to each other. However, a similar tendency as before is present: For small neighborhood sizes (until $k = 10$) the curve of kernel t-SNE is higher while for larger ones the quality of parametric t-SNE gets better.

The generalization quality of kernel t-SNE on the norb data set (Figure 8, left) is excellent since the quality curves of the training and test set lie very close together. The quality curve of parametric t-SNE for this data set lies much lower. This can be attributed to the fact that parametric t-SNE relies on deep autoencoder networks, for which training constitutes a very critical issue: for an often required large network complexity, a sufficient number of

Table 2: Accuracies of the nearest neighbor classifier for the training and test set of each method on four different data sets.

| data sets | | kernel t-SNE | parametric t-SNE | fisher kernel t-SNE |
|---|---|---|---|---|
| letter | train | 84.1% | 21.3% | 85.5% |
| | test | 80.1% | 27.8% | 80.4% |
| mnist | train | 90.7% | 85.4% | 91.1% |
| | test | 85.8% | 62.5% | 86.3% |
| norb | train | 88.2% | 43.0% | 85.4% |
| | test | 85.4% | 38.5% | 85.6% |
| usps | train | 90.5% | 86.5% | 96.6% |
| | test | 84.8% | 58.6% | 87.4% |

data is necessary for training and valid generalization, unlike kernel t-SNE which, due to it's locality, comes with an inherent strong regularization.

The visualization quality of the usps data set is shown in Figure 8 (right). The quality curves of all methods lie close together, while a similar tendency as previously persists: For small neighborhood sizes the quality of kernel t-SNE is better while for larger values the quality curve of parametric t-SNE is higher.

In many of these evaluations, Fisher kernel t-SNE obtained worse values than kernel t-SNE. This has the following reason: The Fisher metric distorts the original metric (according to the label information) and, therefore, also the neighborhood ranks. However, this is intended since the methods tries to focus on those changes in the data which affect the labeling of the data. Therefore, a better evaluation for this method would be a supervised evaluation like the k-nearest neighbor classifier described in 5. Here, we choose $k = 1$. Table 2 shows the classification accuracy of the visualizations of all data sets and all methods. Here, 'train' refers to the training set of the dimensionality reduction mapping and 'test' to its out-of-sample extension.

This evaluation shows that Fisher kernel t-SNE emphasizes the class structure of the data: The classification accuracies on the out-of-sample extensions are at least as good as those from the other methods. For usps, the accuracy is much better and, therefore, improves the generalization of kernel t-SNE.

## 7 Discussion

We have introduced kernel t-SNE as an efficient way to accompany t-SNE with a parametric mapping. We demonstrated the capacity of kernel t-SNE when faced with large data sets, yielding convincing visualizations in linear time if sufficient information is available in the data set or provided to the method in the form of auxiliary information. For the latter, Fisher kernel t-SNE yields a particularly simple possibility of its integration since the training set can easily be shaped according to the given information.

This proposal opens the way towards life-long or online visualization techniques since the mapping provides a memory of already seen information. It is the subject of future work to test suitability of this approach in stationary as well as non stationary online visualization tasks. Furthermore, it might be beneficial to dynamically adapt the sampled subset $X'$ in order to further improve the generalization towards new data.

## Acknowledgement

This work has been supported by the DFG under grant HA 2719/7-1, by the BMBF within the Leading-Edge Cluster it's OWL and by the CITEC center of excellence.

## References

[1] B. Arnonkijpanich, A. Hasenfuss, and B. Hammer. Local matrix learning in clustering and applications for manifold visualization. volume 23 of *Neural Networks*, pages 476–486. Elsevier, 2010.

[2] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12:2385–2404, 2000.

[3] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2003.

[4] C. M. Bishop, M. SvensÃ©n, and C. K. I. Williams. Gtm: The generative topographic mapping. *Neural Computation*, 10:215–234, 1998.

[5] M. Brand and M. Brand. Charting a manifold. In *Advances in Neural Information Processing Systems 15*, pages 961–968. MIT Press, 2003.

[6] K. Bunte, M. Biehl, and B. Hammer. A general framework for dimensionality reducing data visualization mapping. *Neural Computation*, 24(3):771–804, 2012.

[7] D.Cohn. Informed projections. In S. Becker, S. Thrun, and K. Obermayer, editors, *NIPS*, pages 849–856. MIT Press, 2003.

[8] A. Gisbrecht, D. Hofmann, and B. Hammer. Discriminative dimensionality reduction mappings. In J. Hollmén, F. Klawonn, and A. Tucker, editors, *Advances in Intelligent*

*Data Analysis XI - 11th International Symposium, IDA 2012, Helsinki, Finland, October 25-27, 2012. Proceedings*, volume 7619 of *Lecture Notes in Computer Science*, pages 126–138. Springer, 2012.

[9] A. Gisbrecht, W. Lueks, B. Mokbel, and B. Hammer. Out-of-sample kernel extensions for nonparametric dimensionality reduction. In *ESANN 2012*, pages 531–536, 2012.

[10] A. Gisbrecht, A. Schulz, and B. Hammer. Parametric nonlinear dimensionality reduction using kernel t-sne. *Neurocomputing*, 147:71 – 82, January 2015. Advances in Self-Organizing Maps Subtitle of the special issue: Selected Papers from the Workshop on Self-Organizing Maps 2012 (WSOM 2012).

[11] B. Hammer, A. Gisbrecht, A. Hasenfuss, B. Mokbel, F. M. Schleif, and X. Zhu. Topographic mapping of dissimilarity data. In J. Laaksonen and T. Honkela, editors, *Advances in Self-Organizing Maps, WSOM 2011*, Lecture Notes in Computer Science 6731, pages 1–15. Springer, 2011.

[12] B. Hammer and A. Hasenfuss. Topographic mapping of large dissimilarity datasets. *Neural Computation*, 22(9):2229–2284, 2010.

[13] T. W. House. Obama administration unveils "big data" initiative:announces $200 million in new r&d investments.

[14] S. Kaski, J. Sinkkonen, and J. Peltonen. Bankruptcy analysis with self-organizing maps in learning metrics. *IEEE Transactions on Neural Networks*, 12:936–947, 2001.

[15] T. Kohonen. *Self-Organizing Maps.* Springer, 2000.

[16] J. Lee and M. Verleysen. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing*, 72(7-9):1431–1443, 2009.

[17] J. A. Lee and M. Verleysen. *Nonlinear dimensionality redcution.* Springer, 2007.

[18] J. A. Lee and M. Verleysen. Scale-independent quality criteria for dimensionality reduction. *Pattern Recognition Letters*, 31:2248–2257, 2010.

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

[20] B. Mokbel, W. Lueks, A. Gisbrecht, M. Biehl, and B. Hammer. Visualizing the quality of dimensionality reduction. In M. Verleysen, editor, *ESANN 2012*, pages 179–184, 2012.

[21] J. Peltonen, A. Klami, and S. Kaski. Improved learning of riemannian metrics for exploratory analysis. *Neural Networks*, 17:1087–1100, 2004.

[22] S. Roweis, L. K. Saul, and G. E. Hinton. Global coordination of local linear models. In *Advances in Neural Information Processing Systems 14*, pages 889–896. MIT Press, 2002.

[23] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE*, 290:2323–2326, 2000.

[24] H. Schulz and S. Behnke. Deep learning - layer-wise learning of feature hierarchies. *KI*, 26(4):357–363, 2012.

[25] S. Sun. Tangent space intrinsic manifold regularization for data representation. In *Proceedings of the 1st IEEE China Summit and International Conference on Signal and Information Processing*, pages 1–5, 2013.

[26] J. Tenenbaum, V. da Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[27] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.

[28] W. Torgerson. *Theory and Methods of Scaling.* Wiley, 1958.

[29] L. van der Maaten. Learning a parametric embedding by preserving local structure. *Journal of Machine Learning Research*, 5:384–391, 2009.

[30] L. van der Maaten. Barnes-hut-sne. *CoRR*, abs/1301.3342, 2013.

[31] L. van der Maaten and G. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[32] L. van der Maaten, E. Postma, and H. van den Herik. Dimensionality reduction: A comparative review. Technical report, Tilburg University Technical Report, TiCC-TR 2009-005, 2009.

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

[34] M. Ward, G. Grinstein, and D. A. Keim. *Interactive Data Visualization: Foundations, Techniques, and Application.* A. K. Peters, Ltd, 2010.

[35] K. Weinberger and L. K. Saul. An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1683–1686, Boston, MA, 2006.

[36] Z. Yang, J. Peltonen, and S. Kaski. Scalable optimization of neighbor embedding for visualization. In S. Dasgupta and D. Mcallester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 127–135. JMLR Workshop and Conference Proceedings, May 2013.

[37] H. Yin. On the equivalence between kernel self-organising maps and self-organising mixture density networks. *Neural Networks*, 19(6-7):780–784, 2006.

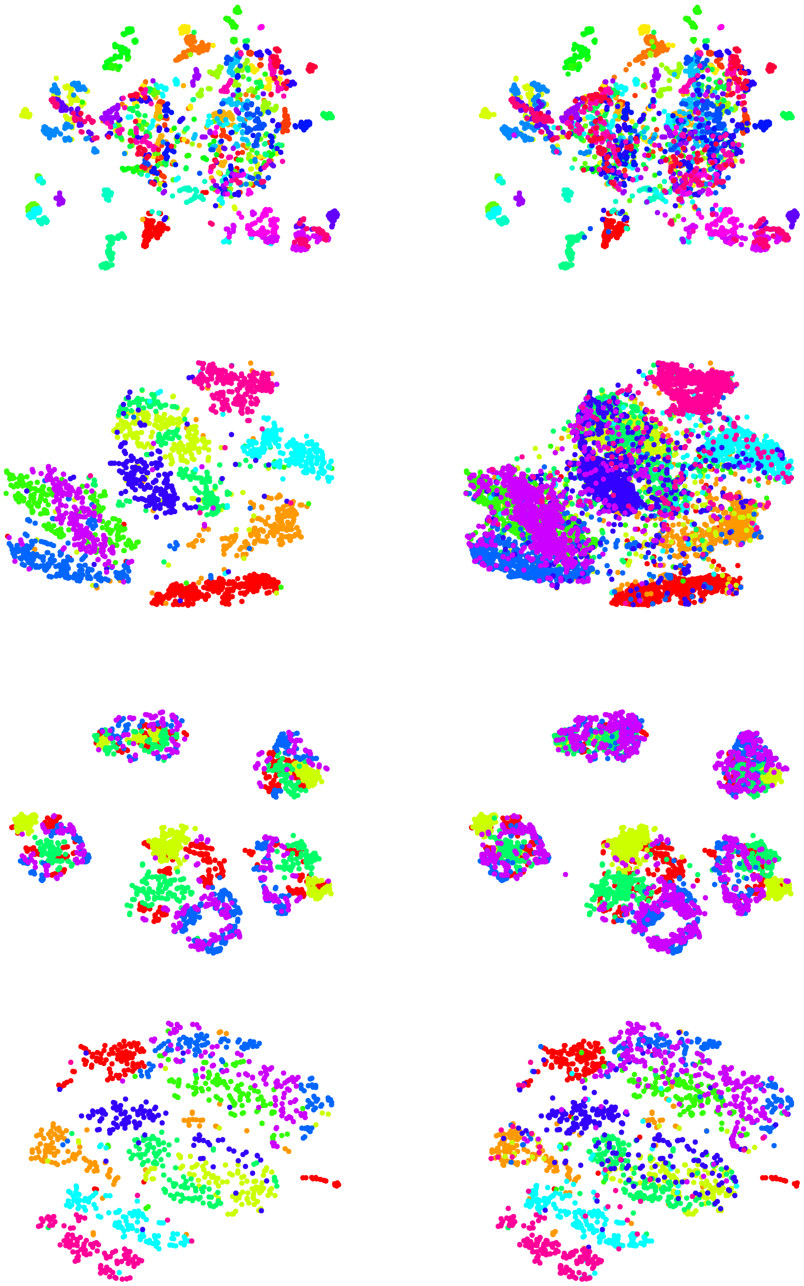

Figure 4: Left column: t-SNE applied on the four data sets letter, mnist, norb and usps (from top to bottom). Right column: out-of-sample extension by kernel t-SNE.

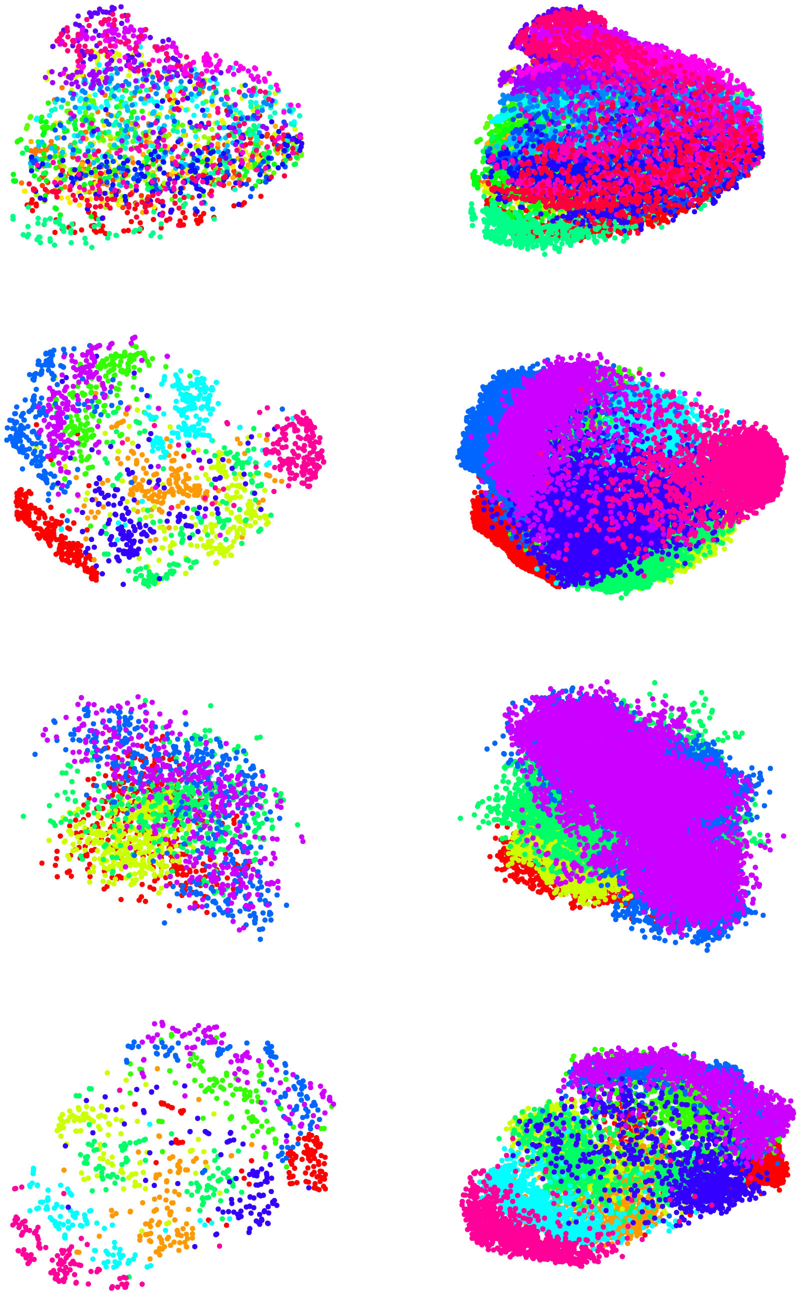

Figure 5: Left column: parametric t-SNE mapping learned from the four data sets letter, mnist, norb and usps (from top to bottom). Right column: out-of-sample extension by parametric t-SNE.

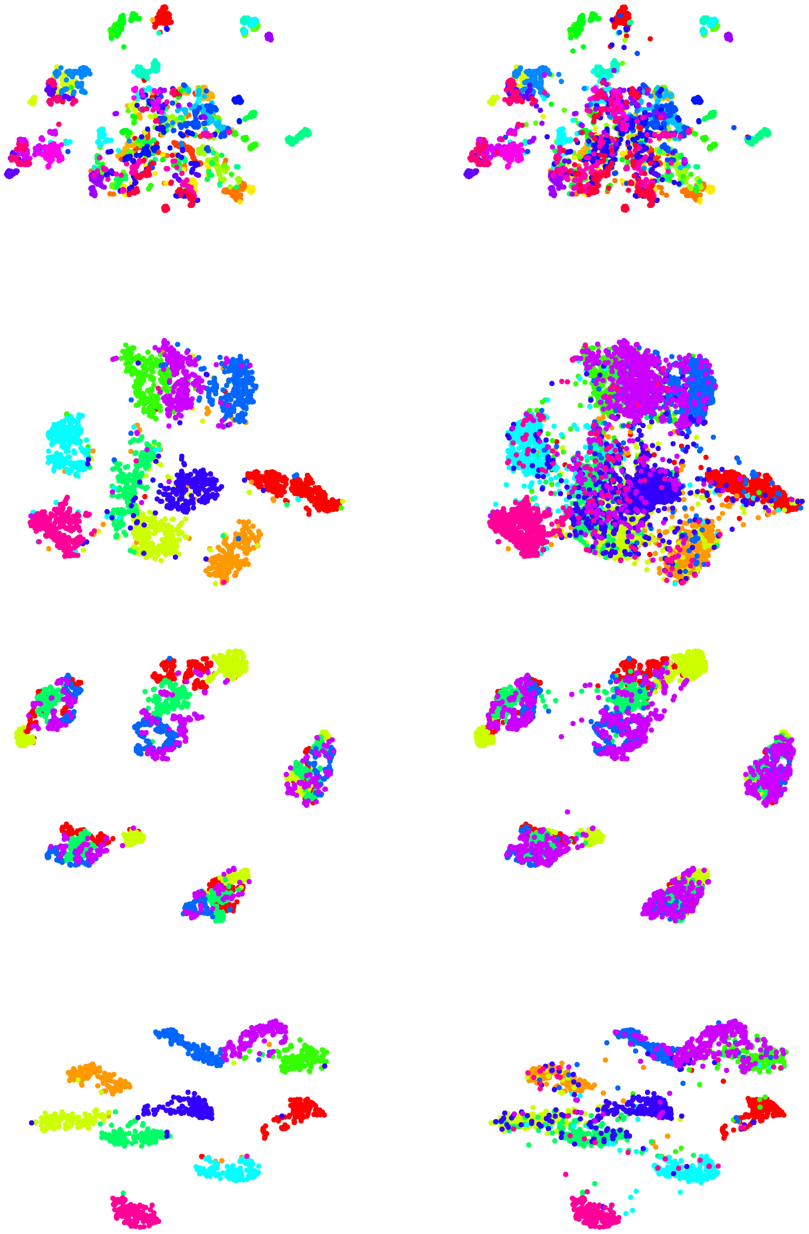

Figure 6: Left column: Fisher t-SNE trained on the four data sets letter, mnist, norb and usps (from top to bottom). Right column: out-of-sample extension by Fisher kernel t-SNE.

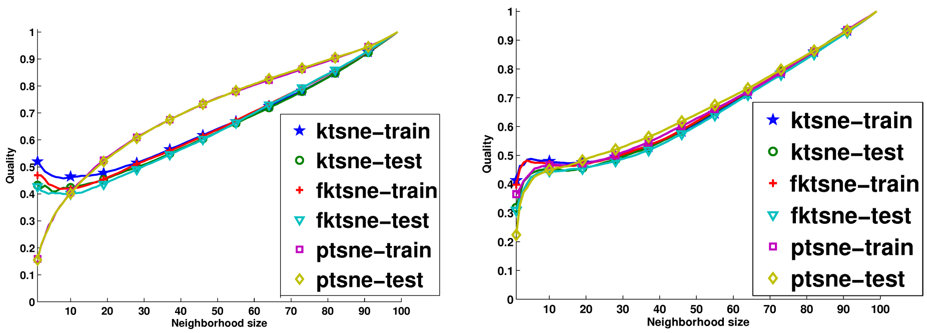

Figure 7: Quality curves for the data sets letter (left) and mnist (right).

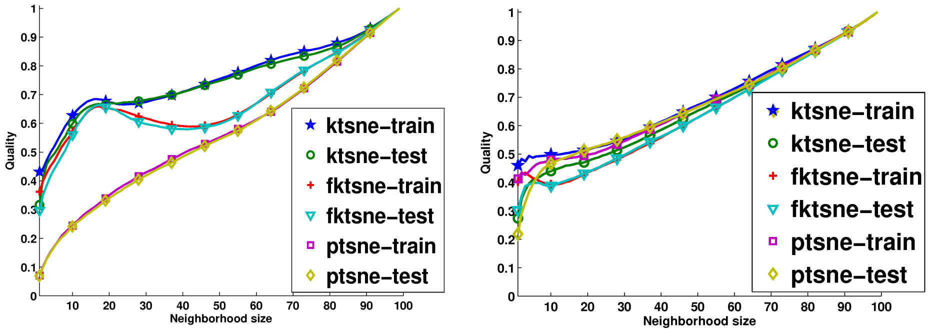

Figure 8: Quality curves for the data sets norb (left) and usps (right).