Efficient Kernelisation of Discriminative Dimensionality Reduction

Alexander Schulz Johannes Brinkrolf
Barbara Hammer
CITEC centre of excellence, Bielefeld University, Germany

Preprint of the publication [29], as provided by the authors.

Abstract

Modern nonlinear dimensionality reduction (DR) techniques project high dimensional data to low dimensions for their visual inspection. Provided the intrinsic data dimensionality is larger than two, DR necessarily faces information loss and the problem becomes ill-posed. Discriminative dimensionality reduction (DiDi) offers one intuitive way to reduce this ambiguity: it allows a practitioner to identify what is relevant and what should be regarded as noise by means of intuitive auxiliary information such as class labels. One powerful DiDi method relies on a change of the data metric based on the Fisher information. This technique has been presented for vectorial data so far. The aim of this contribution is to extend the technique to more general data structures which are characterised in terms of pairwise similarities only by means of a kernelisation. We demonstrate that a computation of the Fisher metric is possible in kernel space, and that it can efficiently be integrated into modern DR technologies such as t-SNE or faster Barnes-Hut-SNE. We demonstrate the performance of the approach in a variety of benchmarks.

1 Introduction

Digital data sets are increasing rapidly as regards size as well as dimensionality. Hence technical support, which enables humans to inspect such data intuitively, becomes indispensable. Besides fully automated classification and data mining, interactive data visualisation plays a prominent role in the context of intelligent data analysis: it enables an inference of hypotheses generation and initial explorative data analysis in the case of complex heterogeneous settings [15, 27, 2].

Nonlinear dimensionality reduction (DR) embeds a given high-dimensional data set into low dimensions, this way enabling a direct visual inspection of the overall structure of the given data set in the form of a scatter plot. In this display, phenomena such as grouping, outliers, or any relevant overall topological structure can be spotted intuitively. Modern DR techniques have enabled striking applications e.g. in biomedical data analysis [5, 14, 17, 24, 32, 35].

Many modern DR methods are phrased as non-parametric techniques [8]; this allows a high degree of nonlinearity when projecting the data, since no prior parametric form restricts the degree of nonlinearity of the overall mapping. While this nonlinearity constitutes a crucial prerequisite for their success, their high flexibility causes the risk to display spurious aspects of the data rather than relevant information especially for high-dimensional or noisy data. Different prominent DR technologies provide quite different visual displays depending on their respective mathematical objective [8]. In general, DR constitutes an ill-posed problem whenever data dimensionality is higher than the dimensionality of the projection space; correspondingly, the results of DR technologies severely differ depending on the used method and even partially depending on its parameterisation.

Discriminative dimensionality reduction (DiDi) offers a very intuitive way to regularise DR technology: in DiDi methods, explicit auxiliary information stratifies in an intuitive way, which aspects of the data are regarded as relevant and which aspects can be discarded when projecting the data to low dimensionality. Technically, a practitioner specifies auxiliary information such as class labels; then DiDi methods subtract all information irrelevant to those aspects from the visual display. This effectively exchanges the original data representation by an alternative one where all aspects irrelevant to the auxiliary labels are divided out. In consequence, this new representation relates to a lower dimensional intrinsic data dimensionality, since it abstracts from quite a few irrelevant aspects. Hence the DR problem becomes more well-posed for this new setting. The result enables an answer to crucial questions as regards the interrelation of data and relevant class labels, such as the following: Do data include any information which relates to the given classes?

Does the data representation offer enough information to robustly separate these classes? Do there exist mis-labellings in the data? Note that, due to its explorative character, DiDi is quite different from data classification itself. It answers the question whether a classification is possible and where potential problems are located rather than inferring a classification itself. Interestingly, DiDi technology can be extended to a full classifier visualisation framework [30].

One particularly powerful general DiDi technology is based on the Riemannian tensor induced by the local Fisher information matrix [10, 23, 26]. The underlying idea is to change the Euclidean metric of the data manifold locally such that feature dimensions which are relevant for the given labelling are emphasised. This defines a Riemannian tensor on the data manifold, hence it induces a Riemannian metric which preserves the underlying Euclidean manifold structure only insofar as it affects the labelling. We refer to this metric as Fisher metric in the following. This metric can be integrated in any DR method which operates on distances only, such as t-SNE [33].

Note that the Fisher metric resembles the important topic of metric learning in some way. The latter aims for machine learning models which adapt a metric according to auxiliary information such that nearest neighbour based retrieval becomes more accurate. Some overview articles on metric learning are [4, 16, 36], for example. Unlike the Fisher metric, these methods are typically parametric, and they are not suited for nonlinear dimensionality reduction for data visualisation. Another related topic falls under the umbrella of multiple kernel learning [11]. Here the goal is the adaptation of a similarity measure by means of a suitable linear combination of given (usually simple) kernels. Unlike the proposed as considered in this contribution, the goal is usually an accurate classification or regression rather than data visualisation.

In this article, we will focus on data visualisation by means of discriminative dimensionality reduction based on the Fisher metric. Like most DiDi methods, however, the existing Fisher-t-SNE technology [10] is restricted to vectorial data. It is not applicable whenever complex, non-vectorial data structures are dealt with. In this contribution we investigate an efficient technology to extend this DiDi method to more general, non-vectorial data structures. One particularly successful approach which enables machine learning for more general objects is based on kernels, which constitute the interface between the machine learning method and the possibly complex application domain [20]. On the one hand, there do exist highly effective structure kernels [1]. On the other hand, kernelisation enables a direct processing of a given discrete Gram matrix [7].

In this contribution, we propose an extension of the Fisher metric to a general kernel space, this way enabling powerful DiDi technologies for general data structures which are described in terms of pairwise relations only. For this purpose, we provide an efficient way how to compute the Fisher matrix itself in kernel space. In addition, we investigate how efficient approximations for the computation of the induced Riemannian metric, which have been proposed in the context of its vectorial version [23, 34], can be kernelised. Finally, we integrate the resulting data description into t-SNE as well as Barnes-Hut SNE for visual inspection. We demonstrate the feasibility of the approach for several benchmarks where data are not given in vectorial form, rather a similarity matrix is available only, including complex structured data from the domains of music and java programming.

This article is structured as follows: First, we recall the concept of the Fisher metric and its efficient computation, which are well established in the literature for vectorial data. Then, we demonstrate how it can be extended to a kernel space, facing two problems: the kernelisation of the Fisher metric tensor, and the kernelisation of its extension to distances by means of approximations of path integrals. We explain how the result can be integrated into t-SNE and Barnes-Hut SNE. We demonstrate the performance of the methods by investigating the discriminative power of the resulting Fisher metric in the original data space, the discriminative behaviour of low-dimensional projections, and their visual display. We conclude with a discussion.

2 Fisher metric

Assume data $\mathbf{x}_i \in X$, $i = 1, \dots, p$ are given, which are elements in an input space $X = \mathbb{R}^D$ of dimensionality D. DR is concerned with a projection of these data to low-dimensional counterparts $\mathbf{y}_i = \pi(\mathbf{x}_i) \in Y = \mathbb{R}^d$ where $d \ll D$, typically $d = 2$ for visualisation. This projection should preserve as much information as possible. Provided the intrinsic data dimensionality is larger than d, information loss, however, cannot be prohibited. For DiDi, auxiliary information is given, which allows the practitioner to control in an intuitive form, which types of information loss are acceptable. We assume that auxiliary information takes the form of data labels $c = c(\mathbf{x})$ where c is element of a finite number of class labels $\{1, \dots, C\}$. Note that an extension to continuous labels is easily possible, see e.g. [31]. The goal is to emphasise those aspects of the data $\mathbf{x}$ in the display which are relevant for $c(\mathbf{x})$. A key observation consists in the fact that popular DR methods rely on pairwise distances of data only, i.e. auxiliary information can easily be integrated by changing the metric according to the labels c. This idea yields consistently superior results as compared to other techniques, which combine discriminative information directly with DR technology [34], and it is applicable for a wide range of DR techniques [30]. Hence, we focus our investigations on this approach.

Now we formally define this Fisher metric. As a first step, a Riemannian curvature tensor is defined. This constitutes the most common way to express the curvature of a data manifold. In our case, this curvature follows the information as contained in the class labels. A Riemannian tensor field is given by a mapping which maps a point $\mathbf{x}$ on the data manifold, to a (pseudo-)metric of the tangent space $T_{\mathbf{x}}M$ of the data manifold M at point $\mathbf{x}$. This metric can be characterised by a positive-semidefinite quadratic form. In our case, this is the Fisher information matrix, which quantifies the influence of the dimensions to a given class label:

$$\mathbf{J}(\mathbf{x}) = E_{p(c|\mathbf{x})}\left\{\left(\frac{\partial}{\partial \mathbf{x}} \log p(c|\mathbf{x})\right)\left(\frac{\partial}{\partial \mathbf{x}} \log p(c|\mathbf{x})\right)^\top\right\}, \qquad (1)$$

where $p(c|\mathbf{x})$ denotes the probability of the class information c conditioned on $\mathbf{x}$ and E denotes the expectation w.r.t this distribution. This matrix is a positive semidefinite form.

As a second step, the Riemannian metric induced by this tensor field is defined via minimum path integrals. Intuitively, given points $\mathbf{x}$ and $\tilde{\mathbf{x}}$ on the manifold, their pairwise distance is computed by infinitesimal steps from $\mathbf{x}$ to $\tilde{\mathbf{x}}$. For those, the local form $\mathbf{J}(\mathbf{x})$ defines the distance, since local directions are elements of the tangent space at $\mathbf{x}$. Since the curvature changes along the manifold, the straight line from $\mathbf{x}$ to $\tilde{\mathbf{x}}$ need not be optimum, rather the optimal path has to be searched for. Formally, a differentiable path from $\mathbf{x}$ to $\tilde{\mathbf{x}}$ is a mapping $P : [0,1] \to X$ with start $P(0) = \mathbf{x}$ and end $P(1) = \tilde{\mathbf{x}}$, which is differentiable with respect to t. For every parameter $t \in [0,1]$, the tangent $P'(t) = dP(t)/dt$ is an element of the tangent space $T_{p(t)}M$, hence its length can be evaluated using $J(p(t))$. For a given path P, its length is obtained by integration over infinitesimal pieces

$$\text{length}(P) = \int_0^1 \sqrt{P'(t)^\top \mathbf{J}(P(t))P'(t)}dt \tag{2}$$

The distance between $\mathbf{x}$ and $\tilde{\mathbf{x}}$ is taken as the minimal achievable length

$$d_M(\mathbf{x}, \tilde{\mathbf{x}}) = \inf_P \{\text{length}(P) \mid P \text{ is a differentiable path from } \mathbf{x} \text{ to } \tilde{\mathbf{x}}\} \tag{3}$$

By definition, d_M constitutes a Riemannian metric, hence its results can be integrated into any distance-based DR method such as t-SNE or extensions thereof. For the algorithmic realisation of this idea, two questions occur: (i) How to compute $p(c|\mathbf{x})$? (ii) How to efficiently compute or approximate the minimum integral (2,3)?

Kernel density estimation of the conditional probability

One generic way to obtain an estimate for a conditional probability distribution relies on a non-parametric density estimation by Parzen windows or kernels. This realisation has successfully been used in the approaches [34, 23, 30] for example. We select a subset $S \subset \{\mathbf{x}_1, \dots, \mathbf{x}_p\}$ of the given data, which act as centres for Gaussians for the density estimation. These are combined as an estimate $\hat{p}(c|\mathbf{x})$ of $p(c|\mathbf{x})$ as

$$\hat{p}(c|\mathbf{x}) = \frac{\sum_{\mathbf{x}_i \in S} \delta_{c=c_i} \exp(-0.5\|\mathbf{x} - \mathbf{x}_i\|^2/2\sigma^2)}{\sum_{\mathbf{x}_i \in S} \exp(-0.5\|\mathbf{x} - \mathbf{x}_i\|^2/2\sigma^2)} \tag{4}$$

The bandwidth σ is often determined by a rule of thumb from the data as explained e.g. in [10]. The Fisher matrix of (4) yields the form

$$\mathbf{J}(\mathbf{x}) = E_{\hat{p}(c|\mathbf{x})} \left\{ \mathbf{b}(\mathbf{x}, c)\mathbf{b}(\mathbf{x}, c)^\top \right\} / \sigma^4 \tag{5}$$

where $\mathbf{b}(\mathbf{x}, c) = E_{\xi(i|\mathbf{x},c)}\{\mathbf{x}_i\} - E_{\xi(i|\mathbf{x})}\{\mathbf{x}_i\}$ with empirical expectation E and probability distributions

$$\xi(i|\mathbf{x}, c) = \frac{\delta_{c=c_i} \exp(-0.5\|\mathbf{x} - \mathbf{x}_i\|^2/2\sigma^2)}{\sum_j \delta_{c=c_j} \exp(-0.5\|\mathbf{x} - \mathbf{x}_j\|^2/2\sigma^2)} \tag{6}$$

$$\xi(i|\mathbf{x}) = \frac{\exp(-0.5\|\mathbf{x} - \mathbf{x}_i\|^2/2\sigma^2)}{\sum_j \exp(-0.5\|\mathbf{x} - \mathbf{x}_j\|^2/2\sigma^2)} \tag{7}$$

see e.g. [23]. We would like to point out that, depending on the data dimensionality, a regularisation of the density estimation is crucial to avoid overfitting, i.e. a misleading visual display. This is one of the reasons why we typically choose S as a subset of all data only. In the experiments, we will always complement the results by the baseline which is obtained when permuting the labels, to demonstrate the validity of the approach. The size of S is chosen such that the baseline is as expected. Note that albeit strong convergence guarantees exist for kernel density estimation provided the number of samples is high [38], we face the challenge of a robust density estimation in the context of a limited number of high dimensional data, a regime which requires careful regularisation.

Efficient approximation of the minimum path integral

The computation of the integral (3) is intractable in general. Thus different efficient approximations are commonly used. The article [23] empirically investigates several paradigms in the context of their integration into a visualisation pipeline, and demonstrates that good results can be obtained e.g. with the so-called k-approximation in particular in the context of discriminative dimensionality reduction. Here we explain this technique in more detail, since we aim for a kernelisation of this method in our approach.

All approximations as proposed e.g. in [23] sample points along the manifold and evaluate the path integral along discrete line segments only. Mathematically, given a finite number of k points $\mathbf{x} = \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k = \tilde{\mathbf{x}}$, a path which moves in line segments trough these points is considered. Assuming that the metric tensor changes smoothly, the following approximation is taken

$$\text{length}_{\text{asym}}(\mathbf{x}_1, \ldots, \mathbf{x}_k) = \sum_{i=1}^{k-1} \sqrt{(\mathbf{x}_{i+1} - \mathbf{x}_i)^\top \mathbf{J}(\mathbf{x}_i)\,(\mathbf{x}_{i+1} - \mathbf{x}_i)} \tag{8}$$

which disregards changes of the metric tensor along the line segments. Note that this definition does not necessarily lead to a symmetric form. If a symmetric form is mandatory, as is the case in our setting, we resort to the following variation: we assume an odd number of $2k+1$ points $\mathbf{x} =$

$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{2k+1} = \tilde{\mathbf{x}}$. Then we evaluate symmetrically:

$$\begin{aligned}\text{length}_{\text{sym}}(\mathbf{x}_1, \ldots, \mathbf{x}_{2k+1}) = \sum_{i=1}^{k} \sqrt{(\mathbf{x}_{i+1} - \mathbf{x}_i)^\top \mathbf{J}(\mathbf{x}_i)\,(\mathbf{x}_{i+1} - \mathbf{x}_i)} \\ + \sum_{i=1}^{k} \sqrt{(\mathbf{x}_{2k-i} - \mathbf{x}_{2k-i+1})^\top \mathbf{J}(\mathbf{x}_{2k-i+1})\,(\mathbf{x}_{2k-i} - \mathbf{x}_{2k-i+1})}\end{aligned} \tag{9}$$

Having defined an approximation of the integral (2), the question occurs how to optimise the path (3). The quality depends on the number k of points and the set of considered paths. One prominent approach samples points which are mutually connected by line segments, and then optimises the overall path by taking minimum distances in the resulting neighbourhood graph. Albeit classical algorithms such as Dijsktra's algorithm address this problem, the computational complexity is $\mathcal{O}(n^2 \log n)$ for sparse graphs and $\mathcal{O}(n^3)$ for fully connected graphs, assuming n data points.

Therefore, commonly, a simpler approach is taken, which provides the same quality empirically if combined with data visualisation [23]: The distance of $\mathbf{x}$ and $\tilde{\mathbf{x}}$ is computed via the line segment from $\mathbf{x}$ to $\tilde{\mathbf{x}}$ only, sampling an odd number of points on the line. Formally, this so-called T-point approximation, for an even number T, is defined as

$$d_T(\mathbf{x}, \mathbf{x}') = \text{length}_{\text{sym}}\left(\mathbf{x}, \mathbf{x} + \frac{\mathbf{x}' - \mathbf{x}}{T+1}, \mathbf{x} + 2 \cdot \frac{\mathbf{x}' - \mathbf{x}}{T+1}, \ldots, \mathbf{x} + T \cdot \frac{\mathbf{x}' - \mathbf{x}}{T+1}, \mathbf{x}'\right) \tag{10}$$

Obviously, this approximation has linear time complexity only. It has been experimentally tested that it does not change the results of a subsequent visual display significantly as compared to more fine grained but also more costly alternatives [23]. This can be attributed to the fact that nonlinear dimensionality reduction methods focus on the preservation of local distances; these are not significantly changed when using the T-point approximation instead of a full optimisation scheme due to the fact that every differentiable manifold is locally approximately flat by definition. Further, the paper [23] evaluated the choice of T. They come to the result, that choosing T larger than 5 does not improve the performance significantly. Hence, we employ this value in our experiments. Having computed this distance matrix, we can use any DR technology which is based on pairwise distances only for its visual display.

3 Kernelisation

In this contribution, we aim for an extension of this approach towards data which are characterised in terms of pairwise similarities only, i.e. kernel values. Hence the question occurs whether this DiDi approach can efficiently be kernelised. In the sequel, we derive a method how these computations can be done directly in kernel space without an explicit embedding of the data. Thereby, the correspondence is exact in the sense that the vectorial counterparts are recovered in case the kernel yields the identity, for more general kernels, the vectorial operations are done in the underlying feature space.

We assume that a similarity matrix $\mathbf{K} \in \mathbb{R}^{N\times N}$ is given with N being the number of data points, entries are denoted as k_{ij}. We assume symmetry of $\mathbf{K}$; hence an implicit vectorial embedding exists $\Phi : \mathbf{x} \to \Phi(\mathbf{x})$ such that $k_{ij} = \Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}_j)$ where the inner product is given by an appropriate symmetric bilinear form in some embedding feature space [12]. Provided a kernel is present, this form is positive semidefinite. For general symmetric $\mathbf{K}$, only bilinearity and symmetry can be guaranteed. In the following, we require non-negativity of all computed pairwise similarities to guarantee a valid probability distribution for an estimation of the Fisher matrix. In particular, this covers the case of structure kernels for complex data structures [20]. Now we inspect in detail how to kernelise the parts required to approximate the Fisher metric.

Kernelisation of the approximation of path integrals

Assume two points $\Phi(\mathbf{x}_i)$ and $\Phi(\mathbf{x}_j)$ are given. The question is how to compute $d_T(\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j))$ based on the kernel matrix $\mathbf{K}$ only? We do not explicitly rely on the embedding space, but we make use of the fact that all points of the line from $\Phi(\mathbf{x}_i)$ to $\Phi(\mathbf{x}_j)$ have the form $\Phi(\mathbf{x}(\alpha)) := (1-\alpha)\Phi(\mathbf{x}_i) + \alpha\Phi(\mathbf{x}_j)$ where $\alpha \in \{0, 1/(T+1), 2/(T+1), \ldots, T/(T+1), 1\}$. Hence $d_T(\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j))$ is a sum of terms of the form

$$\sqrt{\left(\frac{\Phi(\mathbf{x}_j) - \Phi(\mathbf{x}_i)}{T+1}\right)^\top \mathbf{J}(\Phi(\mathbf{x}(\alpha))) \left(\frac{\Phi(\mathbf{x}_j) - \Phi(\mathbf{x}_i)}{T+1}\right)} \qquad (11)$$

This has a computational complexity of $\mathcal{O}(T+1) \cdot \mathcal{O}$ (computation of the quadratic form). Further, we face the question how to efficiently kernelise a quadratic form

$$(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))^\top \mathbf{J}(\Phi(\mathbf{x}(\alpha)))(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)) \tag{12}$$

for data $\Phi(\mathbf{x}_i)$, $\Phi(\mathbf{x}_j)$ and their convex combinations $\Phi(\mathbf{x}((\alpha))$.

Kernelisation of the quadratic form

We find

$$\begin{aligned}
&(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))^\top \mathbf{J}(\Phi(\mathbf{x}(\alpha)))(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)) \\
=&(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))^\top E_{\hat{p}(c|\mathbf{x})} \left\{ \mathbf{b}(\Phi(\mathbf{x}(\alpha)), c)\mathbf{b}(\Phi(\mathbf{x}(\alpha)), c)^\top \right\} / \sigma^4 (\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)) \\
=&\frac{1}{\sigma^4} \cdot \sum_c \hat{p}(c|\Phi(\mathbf{x}(\alpha))) \left((\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))^\top \mathbf{b}(\Phi(\mathbf{x}(\alpha)), c) \right)^2 \\
=&\frac{1}{\sigma^4} \cdot \sum_c \hat{p}(c|\Phi(\mathbf{x}(\alpha))) \left(\Phi(\mathbf{x}_i)^\top \mathbf{b}(\Phi(\mathbf{x}(\alpha)), c) - \Phi(\mathbf{x}_j)^\top \mathbf{b}(\Phi(\mathbf{x}(\alpha)), c) \right)^2 \qquad (13)
\end{aligned}$$

where

$$\begin{aligned}
&\Phi(\mathbf{x}_i)^\top \mathbf{b}(\Phi(\mathbf{x}(\alpha)), c) && (14) \\
=&\Phi(\mathbf{x}_i)^\top \sum_l \left(\xi(l|\Phi(\mathbf{x}(\alpha)), c) \cdot \Phi(\mathbf{x}_l) - \xi(l|\Phi(\mathbf{x}(\alpha))) \cdot \Phi(\mathbf{x}_l) \right) \\
=&\sum_l \left(\xi(l|\Phi(\mathbf{x}(\alpha)), c) \cdot \underbrace{\Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}_l)}_{k_{il}} - \xi(l|\Phi(\mathbf{x}(\alpha))) \cdot \underbrace{\Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}_l)}_{k_{il}} \right) && (15)
\end{aligned}$$

where the sum is taken over all support points in S for the density estimation. Hence, this can be kernelised, provided we find kernel expressions for the terms $\hat{p}(c|\Phi(\mathbf{x}(\alpha)))$, $\xi(l|\Phi(\mathbf{x}(\alpha)), c)$, and $\xi(l|\Phi(\mathbf{x}(\alpha)))$. These are combinations of exponential functions, where the exponent depends on $\Phi(\mathbf{x}(\alpha))$ by means of terms of the form

$$\begin{aligned}
&\|\Phi(\mathbf{x}_l) - \Phi(\mathbf{x}(\alpha))\|^2 = \\
&\underbrace{\Phi(\mathbf{x}_l)^2}_{k_{ll}} + \underbrace{\Phi(\mathbf{x}_i)^2}_{k_{ii}} + \underbrace{\Phi(\mathbf{x}_j)^2}_{k_{jj}} - 2\alpha \underbrace{\Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}_l)}_{k_{il}} - 2(1-\alpha) \underbrace{\Phi(\mathbf{x}_j)^\top \Phi(\mathbf{x}_l)}_{k_{jl}} \qquad (16) \\
&+ 2\alpha(1-\alpha) \underbrace{\Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}_j)}_{k_{ij}}
\end{aligned}$$

Hence, the full computation can be kernelised. The complexity to compute this kernelised version of the quadratic form is $C \cdot |S|$.

4 t-SNE and Barnes-Hut SNE

t-distributed stochastic neighbour embedding (t-SNE) and its efficient approximation, Barnes-Hut SNE, have been established as particularly powerful nonlinear DR technology in the last years [33, 32]. The methods are particularly suitable to investigate the presence of cluster structures in high dimensional data. t-SNE relies on probabilities in the original data space $p_{ij} = (p_{(i|j)} + p_{(j|i)})/(2m)$ where

$$p_{j|i} = (\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2))/(\sum_{k\neq i}\exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)) \tag{17}$$

The bandwidth σ_i is determined such that the effective number of neighbours coincides with a priorly specified parameter, the perplexity (which is a robust meta-parameter, typically chosen between 15 and 50). In the projection space, probabilities are induced by the student-t distribution

$$q_{ij} = (1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1})/(\sum_{k\neq l}(1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}) \tag{18}$$

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 optimises this objective by means of a gradient based technique.

While t-SNE provides excellent results, its complexity scales quadratically with the number of data points. Recently, the so-called Barnes-Hut approximation has been introduced (BH t-SNE) [32]. This relies on two ideas: In the data space, $p_{j|i}$ is substituted by a sparse probability matrix, with $p_{j|i} = 0$ if $\mathbf{x}_j$ is not contained in a neighbourhood of $\mathbf{x}_i$. The neighbourhood size is typically chosen as $3 \cdot u$ with the perplexity u of t-SNE. These relevant neighbours can efficiently be computed in averaged time $\mathcal{O}(N \log N)$ for any given metric using a vantage point tree [37].

The t-SNE gradient can be decomposed into two sums

$$4\left(\sum_{j\neq i} p_{ij}q_{ij}Z(\mathbf{y}_i - \mathbf{y}_j) - \sum_{j\neq i} q_{ij}^2 Z(\mathbf{y}_1 - \mathbf{y}_j)\right) \tag{19}$$

where $Z = \sum_{k\neq l}(1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}$. For sparse p_{ij}, the first sum is efficient. The second sum is approximated by exploiting the Barnes-Hut approximation for efficient n-body simulations [3]. Essentially, data in the projection space are arranged along a Quadtree, and sets of points are substituted by only one average of a cell provided the approximation is sufficient. On average, the complexity is also $\mathcal{O}(N \log N)$.

Note that neither t-SNE nor BH t-SNE require data being euclidean. Rather, any metric can be used, such as the Fisher metric. Therefore, we can directly combine the kernel computation of the Fisher metric with these two DR technologies. Note that a vantage point tree does not require the computation of all pairwise distances, rather a quasilinear subset is sufficient. Hence it is advisable to compute kernel values K and corresponding Fisher distances on the fly when constructing the vantage point tree, which results in an overall effort of $\mathcal{O}(N \log N \cdot TC|S|)$, which is quasilinear in the size of the data points, provided the size of the support set S for the kernel density estimation is limited.

5 Experiments

We evaluate the proposed method for six benchmark data sets that are only given as a similarity matrix. Naturally, for vectorial data, the proposed method is identical to its vectorial counterparts provided the identity is used as kernel. Hence we do not evaluate vectorial settings, rather we investigate data which are characterised directly in terms of their pairwise similarity. The data include the following:

Aural Sonar [25]: The data consist of 100 returns from a broadband active sonar system. Their similarity is evaluated by human experts. Two classes (target of interest versus clutter) are distinguished.

Patrol [7]: 241 members of seven patrol units are characterised by (partially faulty) feedback of unit members naming five colleagues each.

Protein [13]: 226 globin proteins are compared based on their evolutionary distances, four classes of different protein families result.

Voting [7, 19]: 435 either republican or democrat candidates are characterised by 16 nominal attributes which characterise the key votes identified by the CQA, the value difference metric is used for comparison.

Java Programs [21, 22]: 64 Java programs which implement bubble sort or insertion sort, respectively, have been retrieved from the internet. They are compiled with the Oracle Java Compiler API and compared by alignment.

Sonatas [9]: 1068 sonatas in MIDI format from the online collection *Kunst der Fuge* are transformed to graph structures and compared with the normalised compression distance of their paths, labelling is given by one of 5 composers from the classical / baroque era.

A more detailed description of the data can be found in [7, 9].

Each data set is characterised in terms of a symmetrised similarity matrix $\mathbf{K}$. Some of the data do not constitute valid kernel matrices, in such cases pre-processing by clipping negative eigenvalues is possible. However, the Fisher metric does not necessarily require a valid kernel, but only that all computed pairwise distances of data points to be non-negative. We perform the following experiments:

- plain t-SNE projection
- projection using Fisher t-SNE
- projection using BH Fisher t-SNE

All data are projected to two dimensions. As discussed before and suggested by the literature [23], the parameter T for the T-point approximation is chosen as 5. The perplexity is set to the default value 20. The bandwidth for the kernel density estimation is chosen as the average bandwidth determined by t-SNE based on the given perplexity. The size of the set S of support points for the kernel density estimation is optimised such that no overfitting occurs in a baseline. In order to have a reference projection, we employ the basic methods kernel PCA [28] which applies the kernel trick to classical PCA, aligning data such that maximum variance is preserved. Further, we compare to the kernel Discriminant Analysis [6] which kernelises classical LDA as a popular linear discriminative projection method. The regularisation parameter of the latter is also tuned such that no overfitting occurs in an according baseline.

We quantitatively evaluate the results by a 1-nearest neighbour (1-NN) classification in the data space using Euclidean, Fisher, and approximated

Table 1: 1-NN classification errors in percent for the investigated data sets

	AuralS	Patrol	Protein	Voting	Java	Sonatas
original data	21	17	77	6	14	13
Fisher metric	8	15	0.9	5	11	10
VP approx.	8	14	9	5	11	17
KPCA	18	28	12	7.6	12.5	26
KDA	15	29	1.4	4	14	20
t-SNE	18	87	31	7	15	10
Fisher t-SNE	11	16	4	4	12	9
BH Fisher t-SNE	12	17	6	5	13	13
baseline KDA	39	76	51	45	49	50
baseline Fis-t-SNE	40	81	48	43	45	49
baseline BH Fis-t-SNE	48	88	59	46	45	39

Fisher metric (which we also refer to as vantage point tree (VP) approximation), and in projection space using t-SNE and BH t-SNE and the standard euclidean metric, respectively. Thereby we also report the result which we obtain when applying Fisher t-SNE to data with randomly permuted labels, which corresponds to the quality which is merely due to statistical effects of the data. We refer to the 1-NN error in this setting as a baseline. Note that it is not reasonable to evaluate the projections by the quality framework for DR evaluation [18] since we do not aim to preserve neighbourhoods based on euclidean distances.

Results are displayed in Tab.1. As can be seen, an integration of the Fisher matrix constantly improves the cluster separability of the display. In all cases, the BH approximation also yields comparative results, enabling a computation in quasilinear time instead of quadratic complexity. As expected because kernel PCA is an unsupervised method, it performs mostly much worse as measured by the nearest neighbour error. Kernel Discriminant Analysis performs competitively on some data sets (on Protein it is even better), but is clearly worse on three others. Here, the reason is the smaller flexibility to follow highly nonlinear data structures as compared to Fisher t-SNE.

The corresponding visualisations are displayed in Figs.1,2,3. It is clearly visible that an integration of the Fisher information leads to a better emphasis of the cluster structures. An approximation with BH Fisher t-SNE mostly preserves this overall impression, whereby there are details which do not coincide with Fisher t-SNE – this is not surprising since the technique

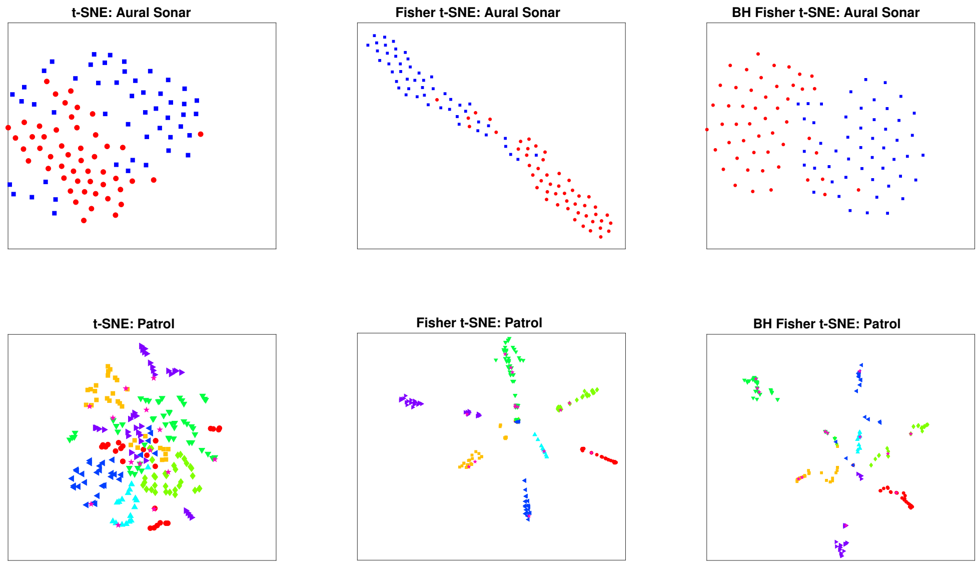

Figure 1: Diverse t-SNE projections for the Aural Sonar data set (upper four figures) and the Patrol data set (lower four figures). Four each quartet, the display shows plain t-SNE (upper left), Fisher t-SNE (upper right), BH Fisher t-SNE (lower right) and Fisher t-SNE applied for randomly permuted labels (lower left).

constitutes an approximation. For all settings, the baseline obtained by random permutation of the class labels yields a widely unstructured image, i.e. structure emphasised by Fisher t-SNE can be attributed to information which is available in the data rather than overfitting effects.

6 Conclusion

In this contribution we have reformulated one particularly popular approach for discriminative dimensionality reduction such that it is applicable to nonvectorial data given by similarities or kernel values. We have shown that the computation of the Fisher metric can be kernelised, and the resulting matrix can be used to drive t-SNE as well as the more efficient BH t-SNE approximation. We evaluated this method with six data sets and obtained a clear improvement as compared to unsupervised projections. The resulting algorithm displays complexity $\mathcal{O}(N \log N)$ provided the set of support vectors

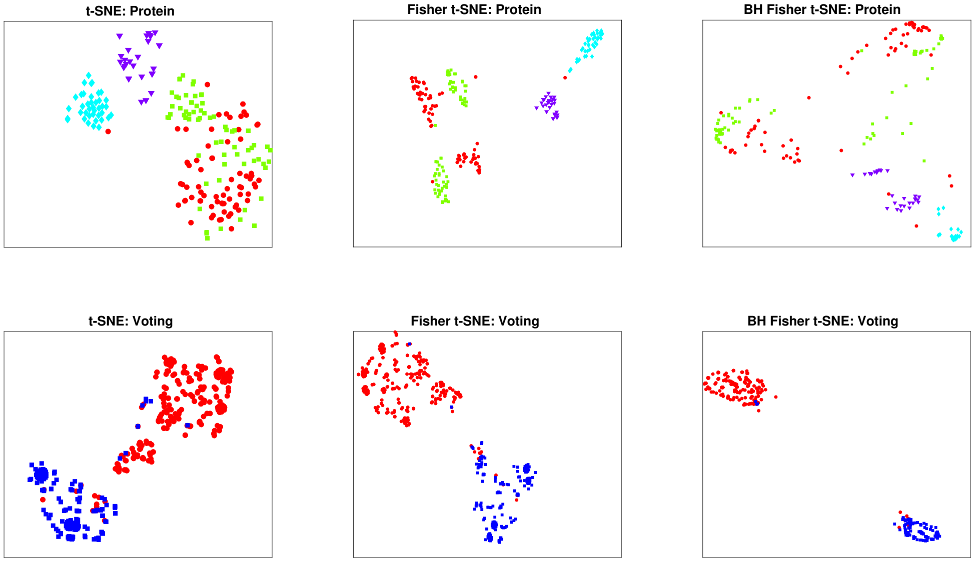

Figure 2: Diverse t-SNE projections for the Protein data set (upper four figures) and the Voting data set (lower four figures). Four each quartet, the display shows plain t-SNE (upper left), Fisher t-SNE (upper right), BH Fisher t-SNE (lower right) and Fisher t-SNE applied for randomly permuted labels (lower left).

for kernel density estimation is chosen of constant size.

The presented framework opens the door to further investigations: the Fisher metric can be extended to auxiliary information which is real-valued by means of a density estimation based on a suitable probabilistic regression model such as a Gaussian process. Since the latter is kernelised, an extension of the proposed framework to real-valued auxiliary information is immediate.

DiDi technology lies at the hart of a classifier visualisation framework as proposed in [30]. It would be interesting to investigate an extension of this framework to kernel values. This would allow practitioners to not only inspect the given data but also a given classifier for complex data structures such as a kernel machine based on structure kernels. This would enable the intuitive interactive exploration of the demanding task of classifier design for structured objects.

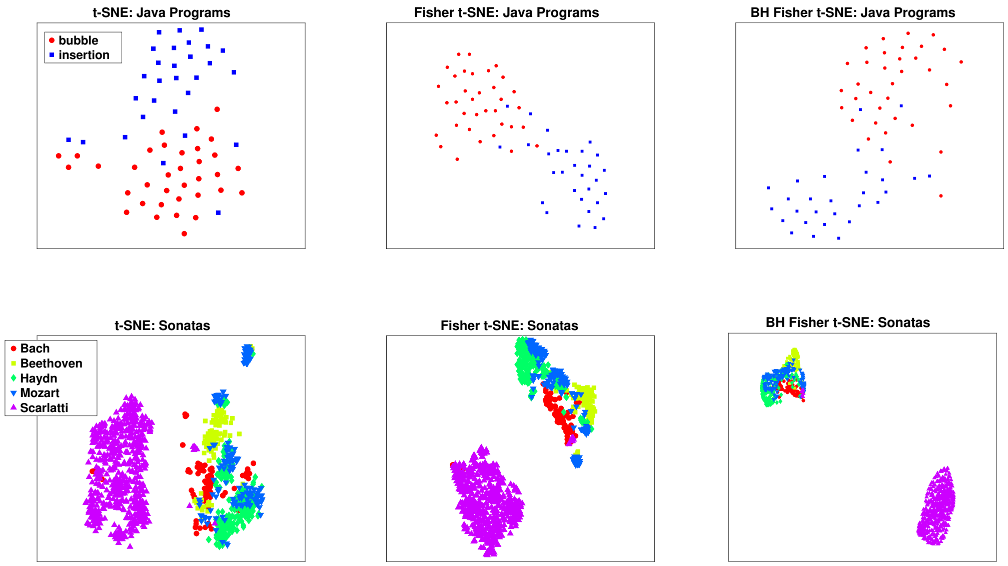

Figure 3: Diverse t-SNE projections for the Java data set (upper four figures) and the Music data set (lower four figures). Four each quartet, the display shows plain t-SNE (upper left), Fisher t-SNE (upper right), BH Fisher t-SNE (lower right) and Fisher t-SNE applied for randomly permuted labels (lower left).

Acknowledgement

Funding from DFG under grant number HA2719/7-1 and by the CITEC center of excellence (EXC277) is gratefully acknowledged. Further, this project is funded by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster Competition its OWL (intelligent technical systems OstWestfalenLippe).

References

[1] F. Aiolli, G. D. S. Martino, and A. Sperduti. An efficient topological distance-based tree kernel. *IEEE Trans. Neural Netw. Learning Syst.*, 26(5):1115–1120, 2015.

[2] M. Aupetit and L. van der Maaten. Introduction to the special issue on

visual analytics using multidimensional projections. *Neurocomputing*, 150:543–545, 2015.

[3] J. Barnes and P. Hut. A hierarchical o(n log n) force-calculation algorithm. *Nature*, 324(6096):446–449, 12 1986.

[4] A. Bellet, A. Habrard, and M. Sebban. *Metric Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2015.

[5] K. Bunte, M. Järvisalo, J. Berg, P. Myllymäki, J. Peltonen, and S. Kaski. Optimal neighborhood preserving visualization by maximum satisfiability. In *AAAI*, pages 1694–1700, 2014.

[6] D. Cai, X. He, and J. Han. Speed up kernel discriminant analysis. *The VLDB Journal*, 20(1):21–33, 2011.

[7] Y. Chen, E. K. Garcia, M. R. Gupta, A. Rahimi, and L. Cazzanti. Similarity-based classification: Concepts and algorithms. *JMLR*, 10:747–776, 2009.

[8] A. Gisbrecht and B. Hammer. Data visualization by nonlinear dimensionality reduction. *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery*, 5(2):51–73, 2015.

[9] A. Gisbrecht, B. Mokbel, and B. Hammer. Relational generative topographic mapping. *Neurocomputing*, 74(9):1359–1371, 2011.

[10] A. Gisbrecht, A. Schulz, and B. Hammer. Parametric nonlinear dimensionality reduction using kernel t-sne. *Neurocomputing*, 147:71–82, 2015.

[11] M. Gönen and E. Alpaydin. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268, 2011.

[12] B. Hammer, D. Hofmann, F. Schleif, and X. Zhu. Learning vector quantization for (dis-)similarities. *Neurocomputing*, 131:43–51, 2014.

[13] T. Hofmann and J. M. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(1):1–14, 1997.

[14] S. Kaski and J. Peltonen. Dimensionality reduction for data visualization [applications corner]. *IEEE Signal Process. Mag.*, 28(2):100–104, 2011.

[15] D. A. Keim and T. Schreck. Special issue on visual analytics. *it - Information Technology*, 57(1):1–2, 2015.

[16] B. Kulis. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364, 2013.

[17] C. C. Laczny, N. Pinel, N. Vlassis, and P. Wilmes. Alignment-free visualization of metagenomic data by nonlinear dimension reduction. *Scientific Reports*, 4:4516 EP –, 03 2014.

[18] J. A. Lee and M. Verleysen. Scale-independent quality criteria for dimensionality reduction. *Pattern Recognition Letters*, 31(14):2248–2257, 2010.

[19] M. Lichman. UCI machine learning repository, 2013.

[20] G. D. S. Martino and A. Sperduti. Mining structured data. *IEEE Comp. Int. Mag.*, 5(1):42–49, 2010.

[21] B. Paaßen. Java Sorting Programs, doi: 10.4119/unibi/2900684, 2016.

[22] B. Paaßen, B. Mokbel, and B. Hammer. Adaptive structure metrics for automated feedback provision in Java programming. In M. Verleysen, editor, *Proceedings of the ESANN*, 2015.

[23] J. Peltonen, A. Klami, and S. Kaski. Improved learning of riemannian metrics for exploratory analysis. *Neural Networks*, 17(8-9):1087–1100, 2004.

[24] D. H. Peluffo-Ordóñez, J. A. Lee, and M. Verleysen. Recent methods for dimensionality reduction: A brief comparative analysis. In *ESANN*, 2014.

[25] S. Philips, J. Pitton, and L. Atlas. Perceptual feature identification for active sonar echoes. In *OCEANS 2006*, pages 1–6, Sept 2006.

[26] H. Ruiz, S. Ortega-Martorell, I. H. Jarman, J. D. Martín-Guerrero, and P. J. G. Lisboa. Constructing similarity networks using the fisher information metric. In *20th European Symposium on Artificial Neural Networks, ESANN 2012, Bruges, Belgium, April 25-27, 2012*, 2012.

[27] T. Ruotsalo, G. Jacucci, P. Myllymäki, and S. Kaski. Interactive intent modeling: information discovery beyond search. *Commun. ACM*, 58(1):86–92, 2015.

[28] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319, July 1998.

[29] A. Schulz, J. Brinkrolf, and B. Hammer. Efficient kernelisation of discriminative dimensionality reduction. *Neurocomputing*, 2017.

[30] A. Schulz, A. Gisbrecht, and B. Hammer. Using discriminative dimensionality reduction to visualize classifiers. *Neural Processing Letters*, 42(1):27–54, 2015.

[31] A. Schulz and B. Hammer. Discriminative dimensionality reduction for regression problems using the fisher metric. In *2015 International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland, July 12-17, 2015*, pages 1–8, 2015.

[32] L. van der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research*, 15(1):3221–3245, 2014.

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

[34] J. Venna, J. Peltonen, K. Nybo, H. Aidos, and S. Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *JMLR-10*, 11:451–490, 2010.

[35] M. Verleysen and J. A. Lee. Nonlinear dimensionality reduction for visualization. In *ICONIP*, pages 617–622, 2013.

[36] O. Walter, R. Haeb-Umbach, B. Mokbel, B. Paaßen, and B. Hammer. Autonomous learning of representations. *KI*, 29(4):339–351, 2015.

[37] P. N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proceedings of the Fourth Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms, 25-27 January 1993, Austin, Texas.*, pages 311–321, 1993.

[38] A. Zanin Zambom and R. Dias. A Review of Kernel Density Estimation with Applications to Econometrics. *ArXiv e-prints*, Dec. 2012.