Advances in dissimilarity-based data visualisation

Andrej Gisbrecht

A thesis presented for the degree of
Doctor of Computer Science

Cognitive Interaction Technology Center of Excellence
Theoretical Computer Science
Bielefeld University
Germany

Gedruckt auf alterungsbeständigem Papier nach DIN-ISO 9706

Abstract

The amount of collected digital information grows with each day. This development is facilitated by advanced technology, allowing for more detailed and more complex measurements. As a result, huge data sets gathered in this way are no longer easily comprehensible for a human person. Dimensionality reduction constitutes an important tool to visualise highly complex data in two dimensions, allowing users to gain insight about the structure of the data.

In this thesis, we give an overview on dimensionality reduction and discuss challenges arising in this context. Moreover, we elaborate in detail the capabilities of dimensionality reduction on two exemplary techniques: *generative topographic mapping*, as a parametric technique, generating a model of the data; and *t-distributed stochastic neighbour embedding*, as a nonparametric projection technique, based on a cost function. As a core contribution of this thesis, we will discuss four major limitations of dimensionality reduction techniques, and present solutions to overcome these problems.
(i) One important requirement for a visualisation technique regards the extensibility to new data: after constructing a mapping for given data, new data should be visualised in a consistent way. While this is readily available for parametric techniques, nonparametric methods are lacking this property, and we present a general approach to provide an explicit mapping.
(ii) Another limitation regards the treatment of complex data: in many scenarios classical feature vectors are no longer sufficient, instead, the use of pairwise data relations in the form of similarity or dissimilarity measures becomes mandatory. Some algorithms are not capable of dealing with dissimilarity data, therefore we develop an approach to solve this problem via an implicit vectorial embedding.
(iii) The techniques working on (dis-)similarity data suffer from high computational and memory complexity, since the matrix with pairwise relations grows quadratically with the number of data points. To drastically reduce the complexity, the Nyström approximation technique can be applied. In this context, we generalize the Nyström method for arbitrary symmetrical (dis-)similarity matrices, and define a universal framework resulting in linear time algorithms.
(iv) Finally, we address the problem of data visualisation being ill-posed, since there are multiple ways to reduce the dimensionality and it is in general not clear which one is the best. Therefore, we follow the metric learning paradigm to focus on dimensions which are important for class separation. We investigate this approach for a parametric technique on the one hand and introduce a general framework for nonparametric techniques on the other hand.

Acknowledgements

I would like to thank my colleagues, my friends and my family for supporting me during this thesis. I would also like to thank Barbara Hammer for being the best supervisor and very kind person overall.

Contents

Chapter 1

Introduction

Motivation

Due to new developments as regards measurement and sensor technology, dedicated data formats, and data storage technology, the amount of data which can be collected and stored electronically increases rapidly in many application domains [73]: prominent examples include next generation sequencing and genomics data, social networks and the web, environmental data e.g. characterising climate conditions, digital images, data from sensor networks, astronomy, extensive digital literature data bases incorporating meta information as well as full texts, etc. [16, 41, 79]. This tremendous growth of digital information and its potentially high social and economical impact has caused the fact that 'big data' has been proclaimed as one of the major challenges of this century [76, 148]. This increasing digitalisation of society bears both, promises and challenges: On the one hand, the analysis of these data sources carries the promise to gain new insights into relevant existing processes and to advance into new areas of research. On the other hand, there are limits on how much objects a human person can track at the same time [22], so that even experts become overwhelmed with unprocessed digital information, e.g. in the form of a table, after a certain size is exceeded. Hence, one major challenge in this context is how to make such amounts of digital data accessible to humans.

One possibility for immediate data access, to overcome the problem of big data sets or high data dimensionality, is by visualising the data, thus relying on the astonishing cognitive capabilities of a human to process visual information [108]: as an example, humans can instantly capture structural forms such as grouping or outliers in visual displays. Hence, data visualisation offers one prominent methodology to enable the interactive analysis of large amounts of digital information. There are different approaches to perform this task, for example, using specially constructed forms such as diagrams, pie charts, graphs, parallel coordinates or similar [85]. Such technology can be combined with suitable interactive strategies, such as investigated e.g. in the field of visual analytics [168]. Another focus in the context of data visualisation centres around the question how to suitably deal with very high dimensional data, where the dimensionality ranges from a few ten up to a few million entries. In this realm, the field of nonlinear dimensionality reduction has emerged as one important area in machine learning [98, 132, 147, 159].

In this thesis, we will focus on data visualisation techniques based on the vehicle of

dimensionality reduction (DR) methods. This means, data are represented as points in a high-dimensional vector space, and the goal is to generate low-dimensional projections of the data in such a way, that as much structure as possible is preserved [98]. Due to the direct visual display, the latter is then directly accessible for humans. Since it is not priorly clear what the term 'structure' refers to, many different instantiations of this principle have been proposed in the literature, ranging from manifold learning [147] up to the preservation of distances or more local structural descriptions of the data [132]. Further, methods differ as to whether data are explicitly given as vectors, or whether such explicit descriptions are not known and data are characterised in the form of pairwise relationships such as distances or similarities [119]. In the latter case, an additional challenge is given by the fact that dissimilarity data requires quadratic storage capacity, since pairwise relationships increase quadratically with the number of given data points.

The increasing complexity of the data does not only cause novel challenges for the practitioner who wants to inspect such data, but the increasing complexity of data and data volume poses new challenges for the involved visualisation algorithms: Diverse areas of application lead to specialised scenarios which require sophisticated techniques to be dealt with. Hence, the challenge arises how to transform DR techniques in such a way that they can incorporate the specific needs required for a given application scenario. Big data sets prohibit their processing with techniques which require more than constant memory or more than linear time. Thus, machine learning techniques have to be adjusted to meet these strict time and memory restrictions. Further, more and more complex structured data arises in areas such as bioinformatics or text processing. These data can often not reliably be represented by vectors, hence visualisation cannot rely on vectorial techniques, rather pairwise problem specific similarities such as e.g. structure metrics have to be taken into account. Further, for big data sets, often severe noise or irrelevant information is encountered, which requires methods which can focus on relevant information for its reliable processing. In this thesis we will develop techniques which are capable of dealing with such tasks.

Novel contributions in this thesis will address the following questions:

- How to extend DR to include prior information? We will present two different possibilities: metric learning by means of relevance learning, and metric adaptation according to given Fisher information.

- How to extend vectorial DR techniques to deal with non-vectorial data described by pairwise dissimilarities only, such as structured data characterised by structure metrics? We will propose an extension of the generative topographic mapping to general dissimilarities.

- How to extend nonparametric methods to an explicit out-of-sample prescription? We will propose a general way how to extend DR methods to an explicit mapping prescription.

- How to enhance the computational efficiency of nonlinear DR techniques? We will investigate the possibility to integrate the Nyström approximation technique into DR methods which are based on pairwise similarities, and we accompany this

linear time approximation by a formal proof of the matrix approximation also for indefinite symmetric matrices, as well as a general framework how to efficiently realise matrix correction techniques such as flip or clip, if required.

Structure and contributions of the thesis

In the thesis, we first present a gentle overview on the most popular DR techniques and their rationale in chapter 2. There, the historical development of DR is depicted, separating the techniques into parametric and nonparametric ones and discussing their advantages and disadvantages. This chapter also illustrates different problems and new concepts which arise in the context of DR. Further, it gives first ideas about approaches to take them into account. Many of these concepts, such as relevance learning, relational data, out-of-sample extension and efficiency are dealt with in the following chapters more thoroughly. Thus, the second chapter lays the ground towards the challenges tackled in this thesis, by providing a structured overview of the relevant underling DR technology, and discussing the open problems dealt within this thesis.

In the third chapter we extend the generative topographic mapping (GTM), as one very reputable generative approach to DR based on a solid statistical modelling, to relevance learning. This changes the focus of GTM to a discriminative one and allows a qualitatively new view on the data, since auxiliary information is taken into account for the DR process.

To extend the applicability of GTM to certain domains, where the data is given only in form of pairwise relations, we present relational GTM in the forth chapter. While leading to an elegant way to directly tackle pairwise similarities or dissimilarities, this extension increases the computational effort of the technique to quadratic complexity, thus turning it infeasible for large data sets.

This problem is addressed in chapter 5, where we give an in-depth discussion on the nature of the relational data and we present a solution to deal with the complexity of relational techniques. The incorporation of the Nyström approximation technique allows to arrive at linear time and constant memory schemes. Interestingly, it is possible to accompany this approximation technique by formal guarantees as to the consistency of the approximation also for general (possibly non-euclidean) (dis-)similarity matrices. Further, the Nyström approximation opens a way to efficiently perform preprocessing steps for similarities or dissimilarities such as frequently used clip or flip operations.

In the last chapter we present a general approach, which provides a parametric mapping for any nonparametric DR technique. This allows to perform out-of-sample extension efficiently and thus decreases the necessary computational effort to visualize large data sets significantly. Another advantage of this approach is, that it allows to efficiently include label information in the form of the Fisher information matrix, which would be too time consuming otherwise.

To summarise, the questions covered in this thesis are depicted schematically in Table 1.1. We will exemplary deal with two popular DR techniques, which represent two different frameworks to realize DR: parametric generative modelling using the generative topographic mapping (GTM), and nonparametric cost function based data projection

Table 1.1: Questions treated in this thesis.

Topic	Relational	Out-of-Sample	Efficiency		Relevance
Technique	Data	Extension	vectorial	relational	Learning
GTM	?	√	√	?	?
t-SNE	√	?	?		?

using t-distributed stochastic neighbour embedding (t-SNE). For these techniques we address the following questions:

- Which type of data can the technique deal with? While t-SNE can handle every type of data characterized by pairwise dissimilarities, we refer to such data as relational data, GTM is restricted to euclidean vectors in its original form similar to many classical machine learning techniques.
- Does the visualization lend itself to direct out-of-sample extensions for new data points? While this is simple for parametric techniques such as GTM which offer an explicit mapping prescription, this is not obvious for nonparametric techniques such as t-SNE, which provide an embedding of the given training data only.
- How efficient are the techniques? While vectorial GTM offers a linear time method, extensions to general dissimilarities, as well as t-SNE, display quadratic complexity which is infeasible for big data sets.
- How to shape the ill-posed problem of DR such that the information, the user is interested in, is displayed? Both, GTM and t-SNE depend on the given data representation and thus provide unsuitable results if this representation contains irrelevant data or noise. One very elegant way to shape the DR according to the user's needs is by the incorporation of auxiliary information, which guides the relevance of the given inputs.

Background of the thesis

As an important and very hot topic, DR is the subject of ongoing research with many scientists across the world dealing with it. During his PhD studies, the author visited several international conferences and workshops, where he had an opportunity to learn from other researchers and to discuss interesting topics with them, which turned out to be beneficial overall. He was also invited to various research seminars at Dagstuhl and at the Max Planck Institute for the Physics of Complex Systems in Dresden. He participated in the Erasmus teacher exchange programme, in which framework he visited Amaury Lendasse at the Aalto University in Espoo, Finland. Another research visit took him to Birmingham, England, where he collaborated with Peter Tino. He also had a very productive discussion on functional analysis with Alexander Grigor'yan from the Bielefeld University in Germany. Last but not least the author is glad to be a part of the

Theoretical Computer Science group headed by Barbara Hammer and positioned at the Cognitive Interaction Technology Center of Excellence in Bielefeld. During this period the author was supported by the DFG under grant number HA 2719/7-1 and by the CITEC center of excellence.

While working on this thesis the author has contributed to many conference and workshop papers as well as to journal articles. The thesis at hand is based on five of these journal articles. This selection can be seen as a round up representation of the work carried out during the PhD, while other articles were written with only a minor contribution of this thesis author, or on related topics, which were not quite fitting in this thesis thematically. The full list of contributions of the author to research in DR can be seen in the following:

Journal Articles

- Andrej Gisbrecht and Barbara Hammer. Relevance learning in generative topographic mapping. *Neurocomputing*, 74(9):1351–1358, 2011.
- Andrej Gisbrecht and Barbara Hammer. Data visualization by nonlinear dimensionality reduction. *WIREs Data Mining and Knowledge Discovery*, 5:51–73, 2015.
- Andrej Gisbrecht, Bassam Mokbel, and Barbara Hammer. Relational generative topographic mapping. *Neurocomputing*, 74(9):1359–1371, 2011.
- Andrej Gisbrecht, Bassam Mokbel, Frank-Michael Schleif, Xibin Zhu, and Barbara Hammer. Linear time relational prototype based learning. *International Journal of Neural Systems*, 22(5):1250021, 2012.
- Andrej Gisbrecht and Frank-Michael Schleif. Metric and non-metric proximity transformations at linear costs. *Neurocomputing*. Submitted.
- Andrej Gisbrecht, Alexander Schulz, and Barbara Hammer. Parametric nonlinear dimensionality reduction using kernel t-SNE. *Neurocomputing*, 147:71–82, 2015.
- Bassam Mokbel, Wouter Lueks, Andrej Gisbrecht, and Barbara Hammer. Visualizing the quality of dimensionality reduction. *Neurocomputing*, 112:109–123, 2013.
- Xibin Zhu, Andrej Gisbrecht, Frank-Michael Schleif, and Barbara Hammer. Approximation techniques for clustering dissimilarity data. *Neurocomputing*, 90:72–84, 2012.

Conference and Workshop Papers

- Andrej Gisbrecht and Barbara Hammer. Relevance learning in generative topographic maps. In M. Verleysen, editor, *ESANN*, pages 387–392. D side, 2010.
- Andrej Gisbrecht, Barbara Hammer, Bassam Mokbel, and Alexander Sczyrba. Nonlinear dimensionality reduction for cluster identification in metagenomic samples. In *IV*, pages 174–179. IEEE, 2013.

- Andrej Gisbrecht, Barbara Hammer, Frank-Michael Schleif, and Xibin Zhu. Accelerating kernel clustering for biomedical data analysis. In *CIBCB*, pages 154–161, 2011.

- Andrej Gisbrecht, Daniela Hofmann, and Barbara Hammer. Discriminative dimensionality reduction mappings. In *IDA*, volume 7619, pages 126–138. Springer, 2012.

- Andrej Gisbrecht, Wouter Lueks, Bassam Mokbel, and Barbara Hammer. Out-of-sample kernel extensions for nonparametric dimensionality reduction. In *ESANN*, pages 531–536, 2012.

- Andrej Gisbrecht, Yoan Miche, Barbara Hammer, and Amaury Lendasse. Visualizing dependencies of spectral features using mutual information. In *ESANN*, pages 573–578, 2013.

- Andrej Gisbrecht, Bassam Mokbel, and Barbara Hammer. The Nyström approximation for relational generative topographic mappings. In *NIPS workshop*, NIPS workshop on challenges of Data Visualization, 2010.

- Andrej Gisbrecht, Bassam Mokbel, and Barbara Hammer. Relational generative topographic map. In M. Verleysen, editor, *ESANN*, pages 277–282. D side, 2010.

- Andrej Gisbrecht, Bassam Mokbel, and Barbara Hammer. Linear basis-function t-SNE for fast nonlinear dimensionality reduction. In *IJCNN*, 2012.

- Andrej Gisbrecht, Bassam Mokbel, Alexander Hasenfuss, and Barbara Hammer. Visualizing dissimilarity data using generative topographic mapping. In R Dillmann, J Beyerer, U.D. Hanebeck, and T. Schulz, editors, *KI*, pages 227–237, 2010.

- Andrej Gisbrecht, Frank-Michael Schleif, Xibin Zhu, and Barbara Hammer. Linear time heuristics for topographic mapping of dissimilarity data. In *IDEAL*, 2011.

- Andrej Gisbrecht, Alexander Schulz, and Barbara Hammer. How to visualize a classifier? In *Workshop NC^2*, pages 73–83. Machine Learning Reports, 2012.

- Andrej Gisbrecht, Dusan Sovilj, Barbara Hammer, and Amaury Lendasse. Relevance learning for time series inspection. In Michel Verleysen, editor, *ESANN*, pages 489–494, 2012.

- Barbara Hammer, Andrej Gisbrecht, A. Hasenfuss, Bassam Mokbel, Frank-Michael Schleif, and Xibin Zhu. Topographic mapping of dissimilarity data. In *WSOM*, 2011.

- Barbara Hammer, Andrej Gisbrecht, and Alexander Schulz. How to visualize large data sets? In *WSOM*, 2012.

- Barbara Hammer, Andrej Gisbrecht, and Alexander Schulz. Applications of discriminative dimensionality reduction. In *ICPRAM*, pages 33–41, 2013. Best paper award.
- Daniela Hofmann, Andrej Gisbrecht, and Barbara Hammer. Efficient approximations of kernel robust soft LVQ. In *WSOM*, 2012.
- Bassam Mokbel, Andrej Gisbrecht, and Barbara Hammer. On the effect of clustering on quality assessment measures for dimensionality reduction. In *NIPS workshop*, NIPS workshop on Challenges of Data Visualization, 2010.
- Frank-Michael Schleif and Andrej Gisbrecht. Data analysis of (non-)metric proximities at linear costs. In *SIMBAD*, pages 59–74. Springer, 2013.
- Frank-Michael Schleif, Andrej Gisbrecht, and Barbara Hammer. Accelerating kernel neural gas. In *ICANN*, pages 150–158, 2011.
- Frank-Michael Schleif, Andrej Gisbrecht, and Barbara Hammer. Relevance learning for short high-dimensional time series in the life sciences. In *IJCNN*, pages 1–8, 2012.
- Frank-Michael Schleif, Bassam Mokbel, Andrej Gisbrecht, Leslie Theunissen, Volker Dürr, and Barbara Hammer. Learning relevant time points for time-series data in the life sciences. In *ICANN*, volume 7553, pages 531–539, 2012.
- Frank-Michael Schleif, Xibin Zhu, Andrej Gisbrecht, and Barbara Hammer. Fast approximated relational and kernel clustering. In *ICPR*. IEEE, 2012.
- Alexander Schulz, Andrej Gisbrecht, and Barbara Hammer. Classifier inspection based on different discriminative dimensionality reductions. In *Workshop NC^2*. TR Machine Learning Reports, 2013.
- Alexander Schulz, Andrej Gisbrecht, and Barbara Hammer. Using nonlinear dimensionality reduction to visualize classifiers. In Ignacio Rojas, Gonzalo Joya, and Joan Cabestany, editors, *IWANN*, volume 7902, pages 59–68. Springer, 2013.

This chapter is based on: Andrej Gisbrecht and Barbara Hammer. Data visualization by nonlinear dimensionality reduction. *WIREs Data Mining and Knowledge Discovery*, 5:51–73, 2015.

Chapter 2

Overview on nonlinear dimensionality reduction techniques

In this overview, we focus on DR techniques for data visualization such as in the popular approaches [132, 147, 158], for example. Interestingly, albeit the earliest techniques for DR such as multidimensional scaling date back more than 50 years [153], nonlinear DR techniques constitute a very active field of research with a variety of diverse methods having been proposed in the last years [24, 48, 93, 96, 157, 158, 174]. One of the reasons behind this circumstance is given by the fact that DR constitutes an ill-posed problem: the goal of DR is to reduce dimensionality of the data while preserving as much information as possible; however, it is not clear how *information preservation* should be defined in a precise mathematical way. Often, it depends on the given data set and the situation at hand which information is regarded as relevant, and since generally it is not possible to retain the whole information, we have to specify what exactly we want to preserve. In consequence, different mathematical principles emerged according to different reasonable interpretations of information preservation.

Imagine a sheet of paper, which is essentially two-dimensional; assume this sheet is bent and twisted in three dimensions. If we are not interested in spatial orientation but rather in the information written on the paper, we can represent it in only two dimensions without loosing any relevant aspects. DR techniques which try to achieve this goal are called manifold learners, popular ones being presented in [15, 147], for example. One underlying assumption of these techniques is that the data points lie on a two-dimensional manifold embedded somehow in a high-dimensional space. Then, the algorithms try to detect this manifold and visualize it as a two-dimensional figure.

Often, data sets are not so simple and their intrinsic dimension is larger than two, hence a trustful representation of data in two dimensions is not possible. In such a case we can define a goal, in mathematical terms a cost-function of what we want to preserve. Different techniques aim at different aspects of the data which should be preserved, e.g. multidimensional scaling preserves the distances or t-SNE preserves the neighbourhood structures of the data points [18]. In this overview, we will exemplarily explain different principles based on which DR techniques can be built, and we discuss the properties of the resulting techniques as regards their computational complexity, capability of mapping novel data points, way of data representation, and similar. In addition, we exemplarily

present the result of popular DR techniques on two typical benchmark data sets, and we discuss the different results obtained by the algorithms.

Since many data sets cannot be directly accessed in any way, formal quantitative evaluation criteria constitute a crucial aspect of DR besides the visual inspection of the results. Such criteria allow to indicate to the user whether the displayed data are trustworthy or not. In addition, explicit quantitative criteria constitute indispensable tools to compare different techniques and to automatically optimize parameters of the methods. DR being ill-posed, there cannot exist a definite evaluation measure to judge the quality of such methods. Nevertheless, several formal evaluation criteria have been proposed which measure reasonable characteristics of such mappings such as their degree of neighbourhood preservation. We will give a short overview about different measures which have been proposed in this context.

Finally, we will have a short look at a number of issues which are currently in the focus of research: extensions of vectorial techniques to more general data structures, discriminative DR, and big data sets, respectively. Non-vectorial data arise if complex data are dealt with which possess an additional (usually discrete) structure such as bioinformatics sequences, texts, graphs, or tree structures [61]. In such cases, a natural interface to the data is given by a data-adapted similarity or dissimilarity measure such as e.g. an alignment distance for structures. While many DR techniques rely on distances only, vectorial techniques such as the self-organizing map have to be transferred to this setting.

Discriminative DR refers to one particularly promising technique which allows the user to shape explicitly which information should be regarded as relevant for visualization: enhancing the data by class labels, the aspects of the data which are relevant for the given labels should be visualized. This principle can be integrated into DR techniques in different ways, one possibility being e.g. a metric-based approach, as we will discuss in this overview [45].

Finally, a very important issue concerns dealing with large data sets, which are becoming more and more common in recent time [148]. The problem is, that most nonlinear DR techniques provide quadratic time complexity which becomes prohibitive for large data sets. Different methods have recently been proposed to circumvent this problem and which approximate the original techniques in an appropriate way. We will have a short look at recent approaches in this overview.

Data sets

Before explaining basic principles of different DR techniques, we introduce two data sets which we will use to demonstrate the behaviour of the techniques in the following. In the literature, a variety of different benchmark data sets have become popular to test DR techniques with different underlying characteristics such as intrinsically low-dimensional manifolds or clustered data, see e.g. [158, 159]. We will use two data sets with different characteristics in the following, a simple synthetically generated one and a real word data set:

- **sphere** consists of 1,000 three-dimensional data points sampled uniformly from the surface of a sphere as shown in the Figure 2.1 (left). The position in the 3rd

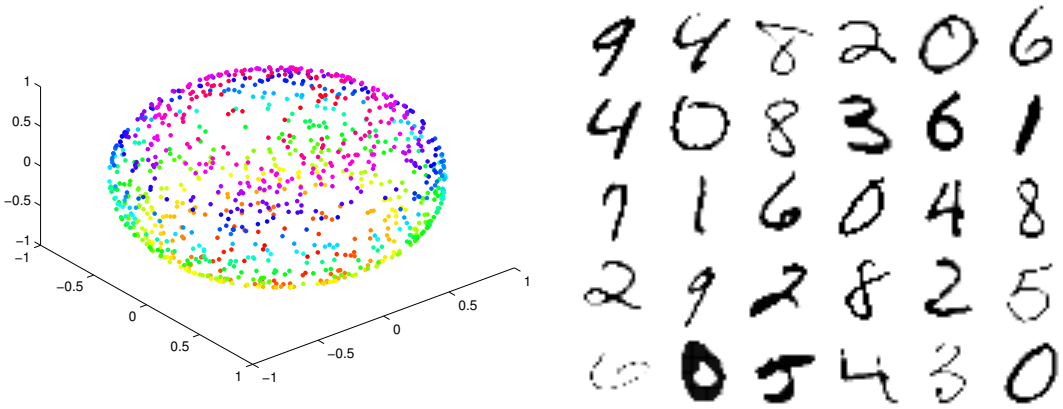


Figure 2.1: Data sets used for the demonstration of the dimensionality reduction techniques, **sphere** (left) and **mnist** (right). For **sphere**, the color is used to represents the position in the 3rd dimension. For **mnist**, examples of handwritten digits are shown.

dimension is also taken as the label of each data point. To better represent the structure of the data set, the label is displayed as a color code in the figures.

- **mnist** is a handwritten digits recognition data set [94]. It consists of 60,000 pictures with 28 times 28 pixels representing the handwritten digits 0 to 9. Each picture can be represented as a 784-dimensional vector. A few digits are exemplarily depicted in Figure 2.1 (right). Since most DR techniques display a high computational complexity, we will use for this data set only a randomly sampled subset of 100 points per class, i.e. 1,000 points, unless stated otherwise.

2.1 Parametric techniques

A principled distinction of different DR techniques can be based on the fact whether they provide a parametric mapping of data points to a low-dimensional embedding, or whether they are nonparametric in nature. Essentially, this distinction addresses the following setting: assume data points $\mathbf{x}$ are given in a high-dimensional data space. The goal of DR techniques is to find corresponding projections $\mathbf{y}$ in two dimensions, such that the characteristics of the projections $\mathbf{y}$ resemble the characteristics of $\mathbf{x}$ as much as possible. For parametric methods, an explicit functional form is taken $\mathbf{y} = f(\mathbf{x})$, one popular choice being a simple linear function f, for example, and parameters of the function f are then determined by the DR technique. Nonparametric methods do not assume a specific functional form, rather they directly assign data points $\mathbf{y}_i$ to every given point $\mathbf{x}_i$ and optimize these projections $\mathbf{y}_i$ directly.

Note that the notion 'parametric' versus 'nonparametric' is not necessarily disjoint and, depending on the given setting, parametric methods can have a nonparametric flavour and vice versa. The essential difference between parametric and nonparametric

methods is that the former are based on an explicit choice of the number of parameters involved which is independent of the number of given data, and an explicit functional form is specified based on this choice. Nonparametric methods choose the number of parameters identical to the number of given data points, and an explicit functional form is not directly given. Hence parametric methods provide out-of-sample extensions per construction while nonparametric methods do not. In practice, there exists a continuum of these methods: methods such as the self-organizing map often use a large number of parameters and the underlying mapping is a simple locally constant, discontinuous mapping provided by the winner takes all rule. Hence, albeit being defined via an explicit functional form, they have a strong nonparametric flavour. Conversely, nonparametric methods can be equipped with interpolation techniques which enhances the methods with an explicit parametric mapping. Albeit this parametric flavour, they are trained without any reference to this mapping, such that we still refer to such settings as nonparametric techniques.

As we will see, most of the classical techniques belong to the parametric class of methods, while many modern nonlinear techniques belong to the nonparametric class, enjoying a larger flexibility since no restriction is imposed on the form of the mapping. However, other drawbacks arise due to this choice, as we will discuss in the following. First, we focus on parametric techniques.

2.1.1 Linear techniques

Linear techniques assume a simple linear functional form of the mapping, thus their flexibility of mapping data is restricted. Principal component analysis (PCA) constitutes a classical technique which is still widely used due to its simplicity and efficiency [153]. It defines a linear mapping

$$\mathbf{y} = \mathbf{L} \cdot \mathbf{x} \tag{2.1}$$

where $\mathbf{x} \in \mathbb{R}^D$ is a point in the high-dimensional data space, $\mathbf{y} \in \mathbb{R}^{D'}$ is a point in the low-dimensional projection space (for visualization, $D' = 2$ holds), and $\mathbf{L}$ is the $D' \times D$ projection matrix. For PCA, the matrix $\mathbf{L}$ is given as the solution of the costs

$$\min_{\mathbf{L}} \sum_i \|\mathbf{x}_i - \mathbf{L}^\top \mathbf{L} \cdot \mathbf{x}_i\|^2$$

for orthonormal vectors in $\mathbf{L}$, where the sum is taken over a finite set of given data points $\mathbf{x}_i$. This cost function models the objective that information of the data should be preserved in the sense that, after projecting the points to low-dimensional space and then back to the high-dimensional space, a point close to the original one should be obtained. Relying on simple algebra, one can show that the rows of the projection matrix correspond to the directions of the largest variance in the data (see Fig. 2.2), thus they correspond to the main principal components of the data space.

As can be seen in Figure 2.3, a linear projection to the largest variances is often not enough to detect relevant structure of the data. In our case, the sphere is projected by simply ignoring one of the three dimensions instead of unfolding it, such that local neighbourhood structures of the sphere become disrupted. Similarly, when considering

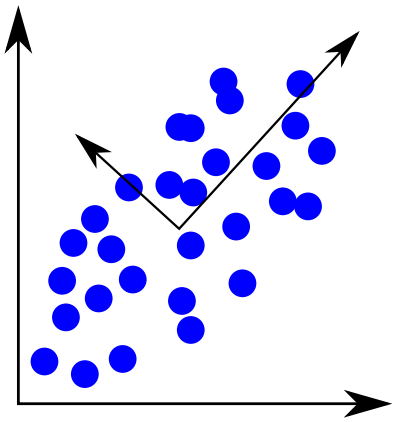


Figure 2.2: The principal components of a two-dimensional data set.

the mnist data set, no separated class structures are visible due to the restriction of PCA to linear mappings only. Still, PCA constitutes a very useful technique for data preprocessing if the dimensionality of a data set is very high, since many directions of the data do not contain any relevant information in such cases. As an example, the 784 dimensions of the mnist data set can be reduced to 50 dimensions by PCA, while preserving nearly all variances present in the data.

Probabilistic formulation of PCA allows to compute the first D' principal components with the time complexity of only $\mathcal{O}(NDD')$ [131], which scales linear in the number of data points. Further, PCA yields a unique projection with an easy interpretation in terms of the data variance. These facts often make PCA the first choice for data inspection, and it can already reveal interesting insights in particular if very high-dimensional data are dealt with [10]. If a data set is extremely high-dimensional, however, even PCA becomes costly.

An alternative widely used linear DR technique is offered by random projection [11, 82]. Similarly to the PCA, the mapping has a linear form where the matrix $\mathbf{L}$ has random values with columns normalized to length one. Random projection constitutes a valid DR technique for high-dimensional data provided the projection dimension is still sufficiently large, in particular, it usually does not constitute a valid visualization method where $D' = 2$ only. The reasons for its validity lie in the Johnson-Lindenstrauss lemma: for such settings the pairwise distances between the given points and their projections are approximately the same with high probability [11].

After a reduction of the dimensionality with PCA it is no longer possible to interpret the features, since they are linear combinations of the original features only. In cases where interpretable features are important as is often the case e.g. in biomedical applications, one can select a small subset of the original features for further computation or also data display. There are different techniques to achieve feature selection [56]. Techniques can roughly be divided into three groups. Filter approaches analyse correlation of features with e.g. Pearson correlation or mutual information to remove redundant features. They are often used as a quick preprocessing technique. Wrapper approaches evaluate the appropriateness of groups of features based on some learning algorithm used to process

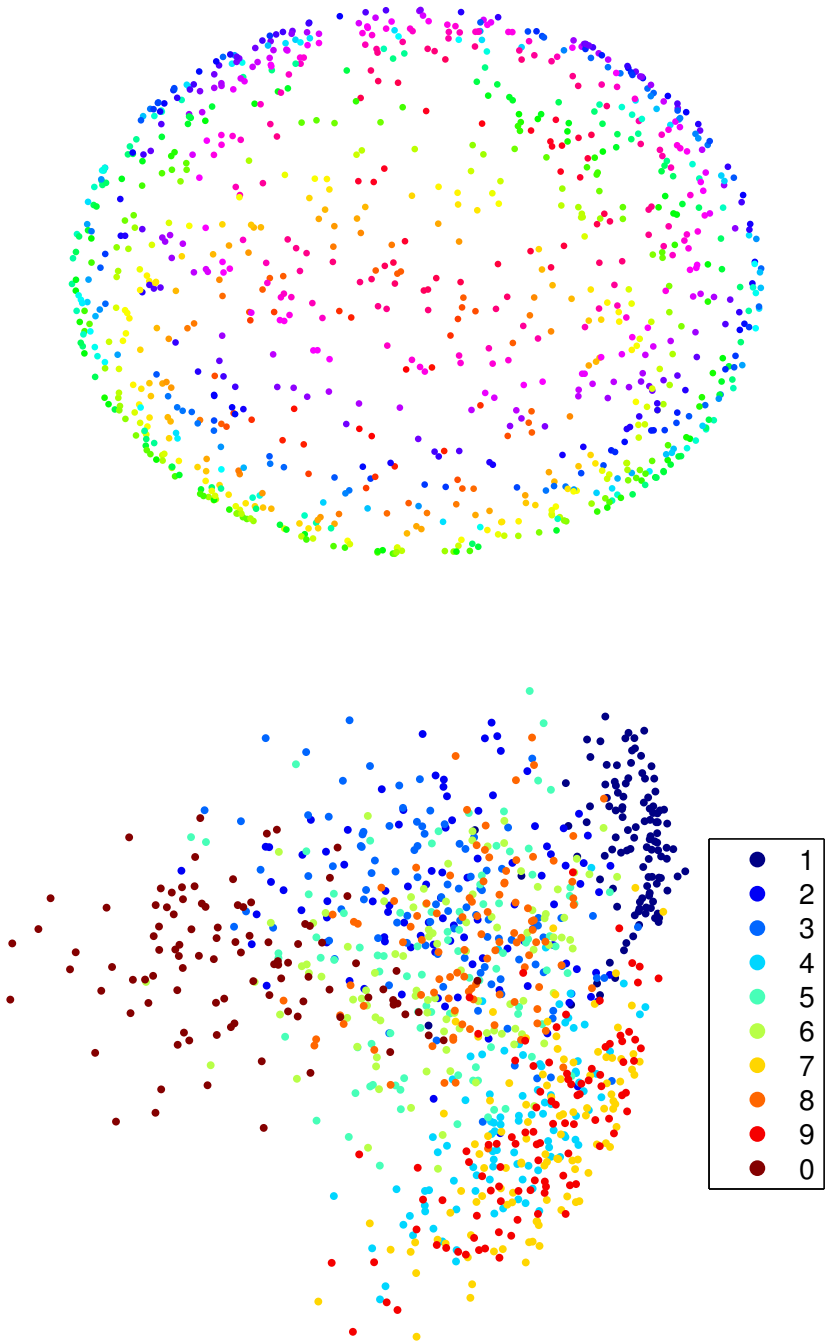


Figure 2.3: PCA projections of the two data sets **sphere** (top) and **mnist** (bottom). The color represents the position in the 3rd dimension for **sphere** and the class for **mnist**.

the data. These feature groups are then modified and the process is repeated until an optimum is reached. For wrapper approaches, it is possible to select relevant features to a specific task. Finally, embedded techniques incorporate the feature selection into a learning algorithm, e.g. by optimizing a cost function which not only maximizes the desired goal, but also minimizes the number of features. For an overview on feature selection see e.g. [56].

2.1.2 Nonlinear extensions of PCA

PCA projects the data onto principal components, i.e. vectors which are essentially straight lines, thus it is limited as regards the data structures it can capture, in particular curved manifolds are beyond its reach. There exists a variety of extensions which transforms linear PCA to more general nonlinear shapes. In principle, these extensions can be differentiated according to two lines: local extensions, where locally linear projections based on PCA are combined, or global extensions where more general shapes than simple lines are considered. In the latter case, the idea is to project the data onto more general shapes, such as curves, manifolds, or even graphs (see e.g. [54, 68]). For manifold charting [15], on the contrary, local fits are found using linear PCA and these charts are then glued together to create a visualization of the whole data set. An alternative way to extend PCA globally to nonlinear projections is by integrating a fixed nonlinear preprocessing or kernel to the mapping. This method, frequently referred to as kernelization, leads to kernel PCA [136]. Thereby, using a kernel function, data are mapped implicitly into a very high-dimensional space where the principal components are computed. Due to this kernel trick it is possible to compute nonlinear projections of data efficiently.

Typical results of these nonlinear extensions of PCA are depicted in the Figures 2.4 and 2.5 for both data sets. Thereby, a Gaussian kernel is used for kernel PCA. Obviously, in all cases, a nonlinear projection is present. Due to its local structure, manifold charting is capable of tearing the original data manifold, providing a continuous projection of the sphere without overlapping regions, and being capable of separating some of the classes of the mnist data sets in the projection. For kernel PCA, the effects are less obvious since it still relies on a globally smooth mapping, and regions where classes are projected on top of each other can still be observed in the projection. Compared to PCA, the computational complexity of the methods increases since kernel PCA requires the computation of the full kernel matrix, which has quadratic effort, while manifold charting requires the coordination of the patches, the computational complexity of which depends on the number of patches and data involved in each of those.

Apart from these extensions, there exists a popular family of techniques which can be interpreted as nonlinear extensions to PCA similar to manifold charting, thereby relying on principles inspired by biological self-organization processes: self organizing maps [176].

2.1.3 Topographic mappings

The self organizing map (SOM) is a neural network motivated by the human brain [86], mimicking self-organizing processes of the cortex. Technically, high-dimensional sensory

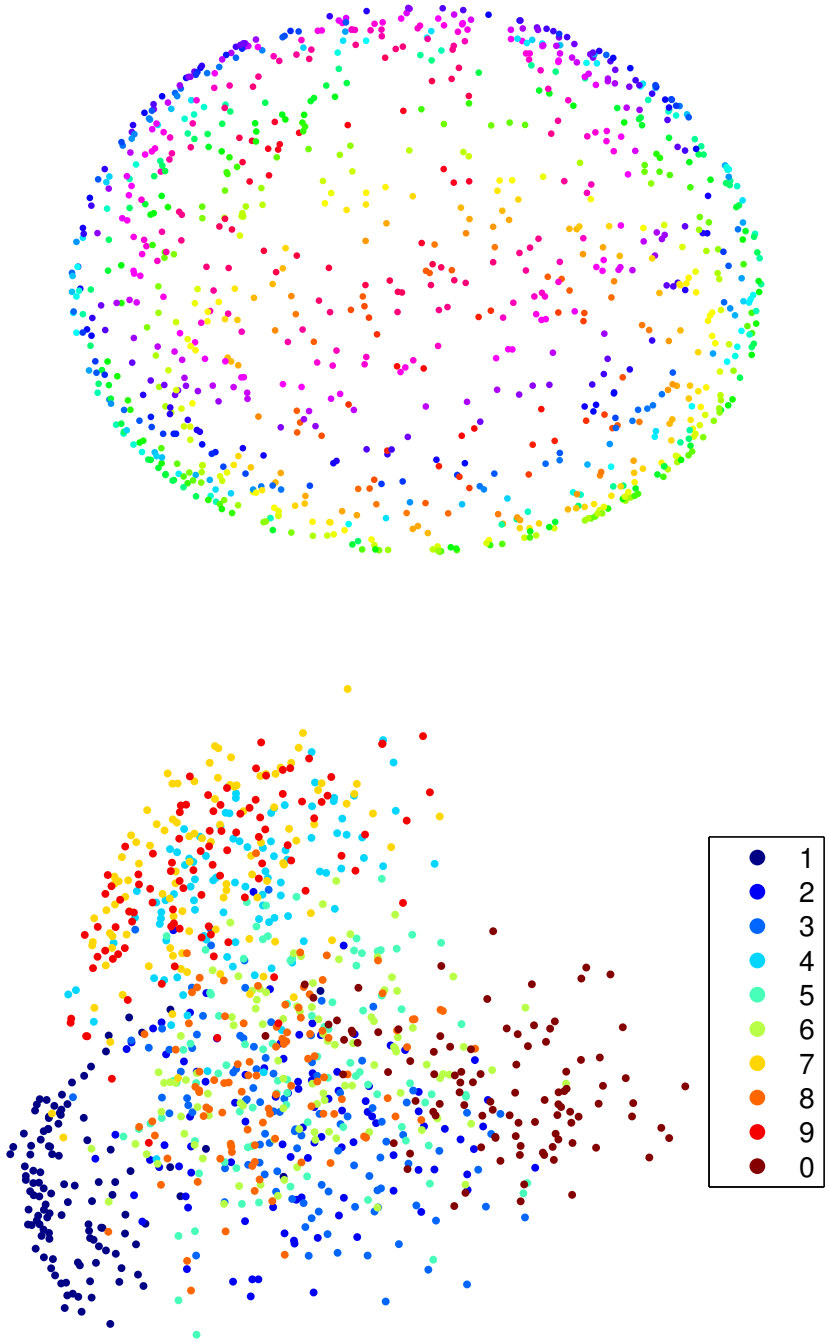


Figure 2.4: Results for kernel PCA for the two data sets **sphere** (top) and **mnist** (bottom). The color represents the position in the 3rd dimension for **sphere** and the class for **mnist**.

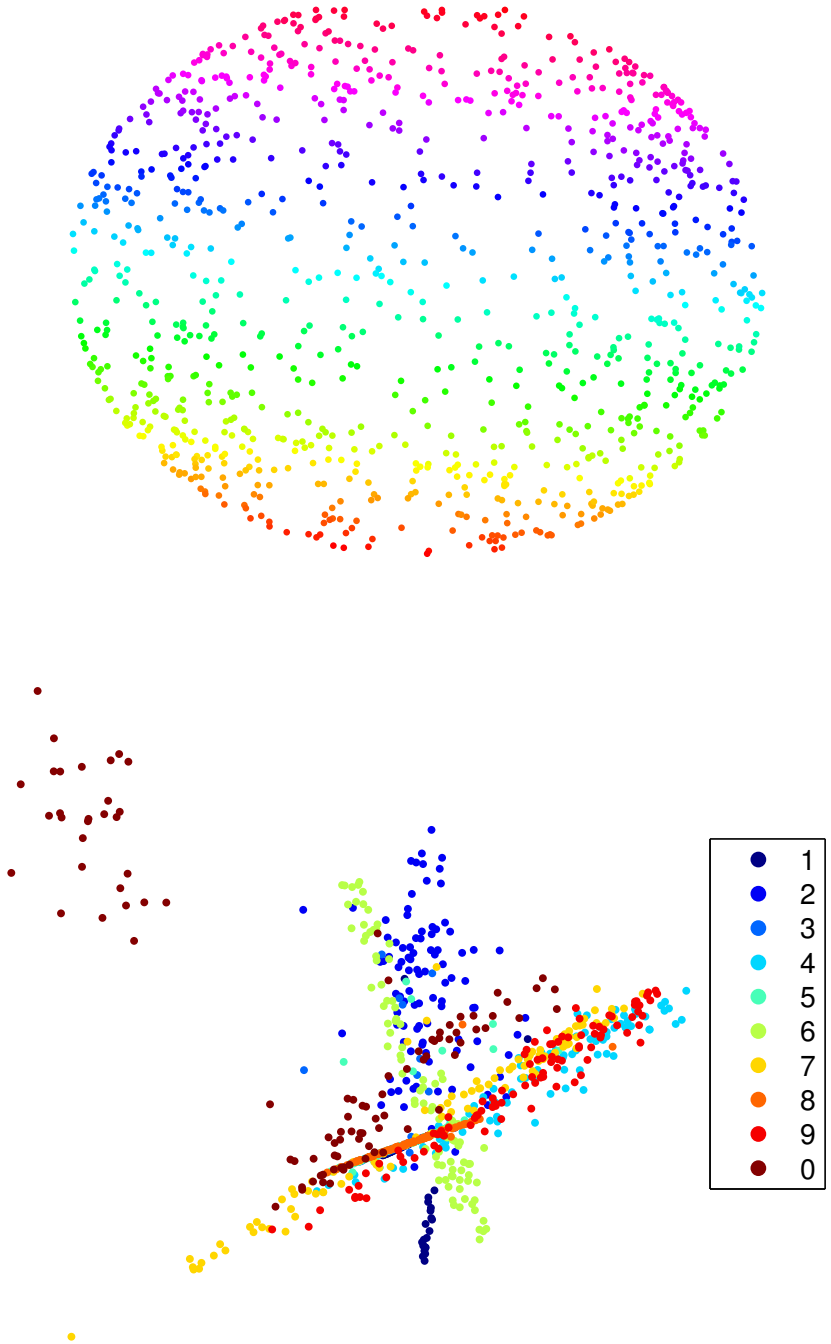


Figure 2.5: Results for manifold charting for the two data sets **sphere** (top) and **mnist** (bottom). The color represents the position in the 3rd dimension for **sphere** and the class for **mnist**.

data are mapped to an essentially two-dimensional manifold in a topology preserving fashion using self-organizing learning.

The SOM consists of neurons which are connected to each other in a regular form in a grid. This could be for example hexagons with the whole network having the form of a honeycomb or, as we will use it in our example, a rectangular grid as depicted in Figure 2.6. Each neuron has an associated weight, which is a vector in the original data space, covering a certain region of the input space, its receptive field. In an online way, the map is trained as follows: a single data point is presented and the winner neuron, i.e. the neuron with the weight closest to this point, is determined. The neuron's weight as well as its neighbourhood in the grid are then pulled towards the direction of the data point. This way, during training, the net unfolds to cover the data and thus can be used for visualization. An example visualization of the sphere data is shown in Figure 2.6. It can be seen that the SOM captures the two-dimensional manifold perfectly, unfolding of it to the original grid yielding a two-dimensional visualization. As explained in [128], for example, SOM can be interpreted as a nonlinear extension of locally linear PCAs. Interestingly, due to its iterative adaptation, SOM has only linear time complexity. Despite its simple training algorithm, its precise mathematical analysis turns out rather complicated. Although it was possible to reformulate SOM as an optimisation of an explicit cost function, still quite a few open problems remain unsolved today [40, 66, 72].

The generative topographic mapping (GTM) constitutes a probabilistic counterpart of SOM [12] which models data in terms of a constraint mixture of Gaussians. In contrast to the original version of the SOM it relies on a cost function of the model which is optimized during training. Similarly to SOM its neurons are placed on a regular grid in a low-dimensional latent space. GTM defines a nonlinear mapping from this latent space to the high-dimensional data space, such that a data distribution in the latent space induces a distribution in the data space, thereby adding Gaussian noise to the grid centres. This way, a low-dimensional manifold given by the grid is embedded in the high-dimensional space. During training, the mapping parameters as well as the bandwidth of the Gaussians are determined in such a way as to maximize the data log likelihood function on an observed set of training data. Usually, training takes place by a classical expectation maximization scheme, leading to a visualization similar to SOM, as shown in Figure 2.7.

We will have a close look at GTM as one prominent parametric DR technique in chapters 3 to 5 where we will discuss the questions how to integrate prior knowledge into GTM, how to extend it to non-vectorial data, and to deal with the resulting quadratic time complexity provided big data sets, respectively.

2.1.4 Auto-encoder network

Auto-encoder networks constitute a DR technique which is based on the principle of encoding and decoding the data as faithfully as possible in low-dimensional space thereby relying on an appropriate nonlinear mapping [75, 159]. Classically, multilayer neural networks have been used for this purpose, whereby an intermediate layer with only few (two) neurons, the encoding layer, provides a visualization of data. In classical approaches, the weights of the connections are trained in such a way that the input

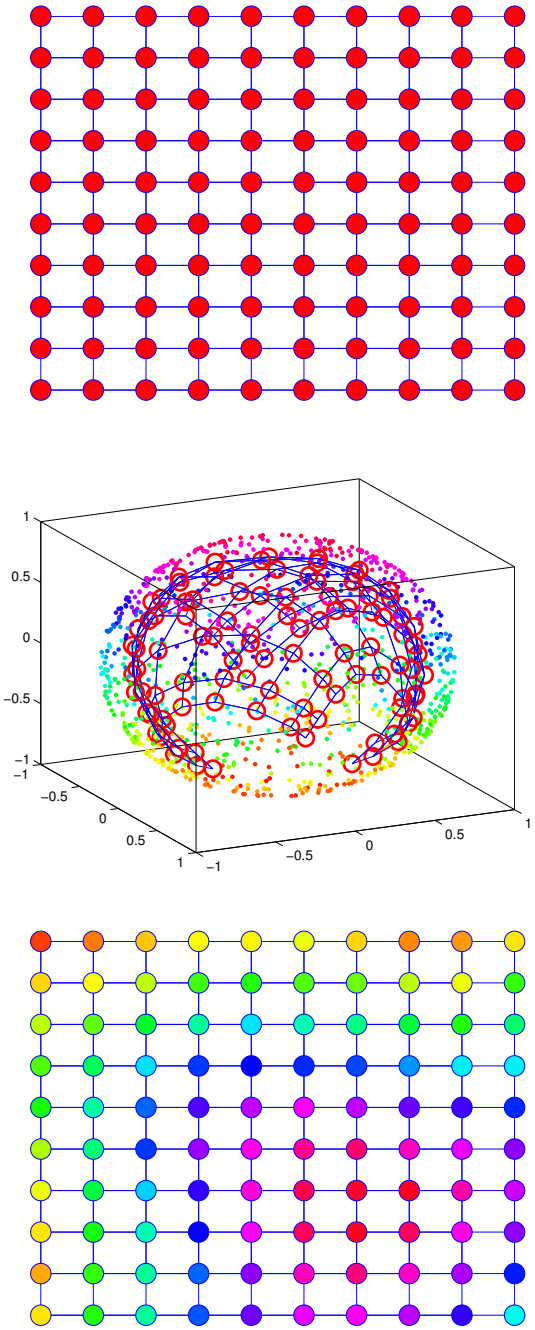


Figure 2.6: Visualization of data using the self-organizing map: (top): a typical rectangular SOM grid, (middle): trained SOM for the sphere data depicted in the data space, (bottom): unfolding of the SOM to achieve a two-dimensional visualization.

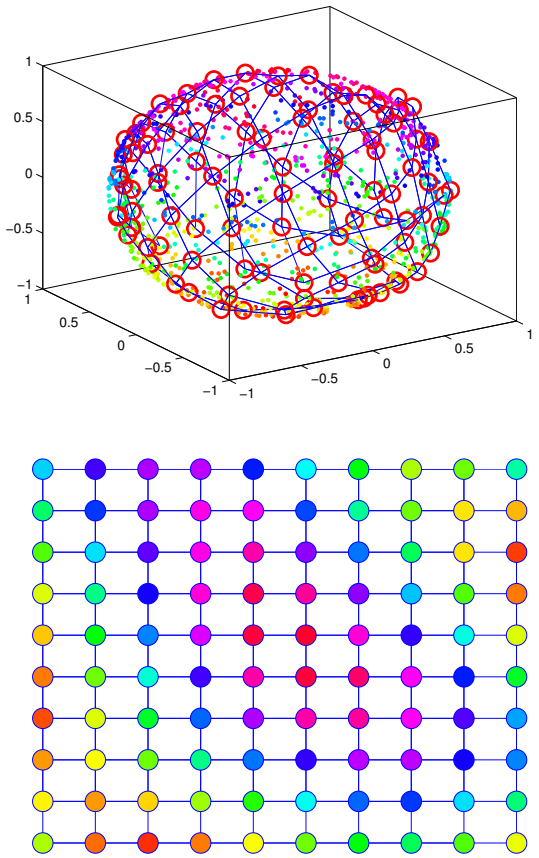


Figure 2.7: Visualization of data using the generative topographic mapping: (top): trained GTM for the sphere data depicted in the data space, (bottom): unfolding of the GTM to achieve a two-dimensional visualization.

vectors are reproduced by the output layer, directly minimizing the mean squared error on a given training set. Although the resulting training complexity is linear in the number of data points, the model has usually a large number of parameters and a deep structure, such that training is complicated.

In recent years, it has been observed that similar deep architectures could be a principle in human information processing of the brain, indicating the cognitive plausibility of this approach [75]. It has been observed that the problems of training deep networks can be avoided by incorporating principles of statistical physics, leading to a new boost of an area typically referred to by deep learning [75, 137]. Due to the typically large number of parameters, however, auto-encoders typically require a large number of training data for valid generalization. The result can be interpreted as a nonlinear extension of PCA with the encoding part of the network to two dimensions yielding two nonlinear components which best describe the data in a least mean squares sense.

All parametric techniques described in this section have in common that they provide an explicit mapping $f(\mathbf{x})$ for the visualization of data, thus the techniques can map new data points per definition. The price paid for this capacity is a limited flexibility of the techniques due to a priorly fixed form of the mapping f. The methods differ in the complexity of f, PCA restricting to very simple linear functions with excellent generalization capability but very limited flexibility, not being capable of capturing nonlinear structures of the given data set. On the contrary, deep auto-encoders built their capability on a flexible mapping given by deep auto-encoder networks, paying for this flexibility with high computational costs and the need for a large number of training data to faithfully determine the model parameters.

The afore-mentioned generalisation capability is important for the parametric as well as nonparametric DR techniques. The process of DR can be seen as learning a regression function from high-dimensional to low-dimensional vectors. Thus, the complexity of the mapping function f should be chosen carefully. A too simple mapping, as e.g. in PCA, would be not sufficient to learn the data structure and a too complex mapping would not trustfully represent the data, as e.g. in a deep auto-encoder network, where each point could be projected to an arbitrary position. The overfitting of a cost function on a small amount of points would result in a mapping where the training points are visualised as best as possible but the regions between these points are mostly ignored, thus disturbing the original data structure. Visualisation of new additional points would generate a completely different projection, or not trustworthy projection of the new points. Both cases are undesired for data analysis.

2.2 Nonparametric techniques

Nonparametric techniques, on the contrary, do not fix a functional form, rather every projection point $\mathbf{y}_i$ can be chosen independently of the others, yielding to a large flexibility of the mappings. This way, nonparametric techniques are capable of following local nonlinear distortions or tears of the data; so they are ideally suited to visualize nonlinear effects of a given data set. As a downside, they do no longer provide an explicit mapping for new data points such that their suitability for DR as preprocessing technique rather

than visualization is limited. Most DR techniques for data visualization which have been proposed in the last years belong to the class of nonparametric techniques.

2.2.1 Multidimensional scaling

Multidimensional scaling (MDS) is the name for a group of classical DR techniques which are based on the aim of distance preservation [98]. Classical MDS (CMDS) or metric MDS tries to preserve the original Euclidean distances of the data in the low-dimensional space. It can be related to PCA by turning pairwise distances into pairwise scalar products. For the resulting similarity matrix, the first two principal components yield the optimum projection dimensions in a least squares sense. CMDS has the benefit of a unique solution which can be determined analytically.

Note that if the data is centred, the PCA and CMDS generate the same results. Their difference comes from the optimised cost function, i.e. while PCA has a priori fixed number of parameters in the cost function, the number of parameters in CMDS is proportional to the number of data points. This makes the former to a parametric technique and the later to a nonparametric one.

Variants of MDS reformulate the cost function of CMDS to account for the fact that the relative relationship or ordering of the neighbourhood should be preserved rather than the exact distances itself. Most variants aim at an optimization of cost functions, typically called stress functions in the context of MDS, which relate to a weighted sum of a costs resulting from differences of the original distances and the distances of projected points to each other, thereby e.g. weighting close points higher than far away pairs for the so-called Sammon's stress. Still, Sammon's mapping often yields restricted results when it comes to highly nonlinear structures, as can be seen in Figure 2.8. Another form of MDS is offered by curvilinear component analysis (CCA) [34], which focusses on distances which are close in the projection space rather than the original data space, weighting these points higher. As CCA, Curvilinear distance analysis (CDA) [95] weights points in low-dimensional space and, as motivated by Isomap, which we will describe in the next section, relies on geodesic distances in high dimensional space. This results in an unfolding of the data manifold, similarly to SOM. Typically, these variations can no longer be optimized in closed form, rather gradient techniques or heuristics are used.

Recently, it has been pointed out that CCA and Sammon's mapping constitute two examples which, in fact, address two different objectives of DR: the trustworthiness of the mapping, referring to the fact that, if two points are displayed close together, this should stem from two points which are also neighboured in the original setting; on the other hand, the continuity of the mapping, referring to the fact that points which are neighbours in the original data space are also depicted as neighbours in the visualization [162]. Based on this observation, local MDS proposes to simultaneously optimize both objectives in a suitably weighted form, such that the user can choose the trade off between trustworthiness and continuity [162].

Due to their intuitive motivation, MDS techniques still constitute very popular nonlinear DR techniques. However, they have the drawback that, similar to PCA, their capability to capture nonlinearities can be rather limited, and non-metric extensions of classical MDS often require the fine-tuning of optimization parameters, since their cost

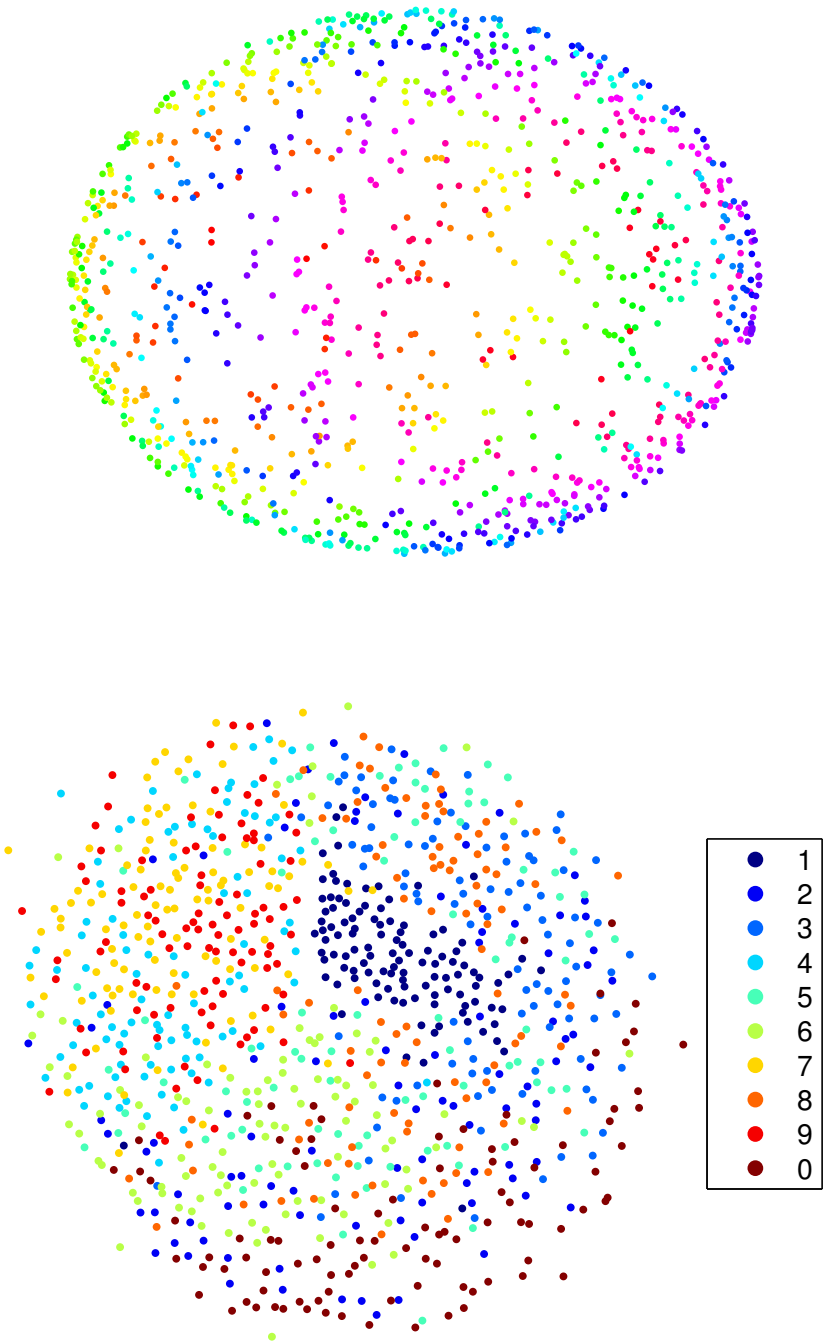


Figure 2.8: Results for Sammon's mapping for the two data sets **sphere** (top) and **mnist** (bottom). The color represents the position in the 3rd dimension for **sphere** and the class for **mnist**.

functions possess local optima. This fact also leads to a non-deterministic behaviour with possibly different visualizations resulting from different runs on the same data set. The computational complexity of the techniques is usually quadratic due to the dependency on pairwise distances.

2.2.2 Spectral embeddings

Spectral embeddings are a group of techniques analysing the data using spectral decomposition, i.e. the eigenvalue decomposition $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, of a similarity matrix $\mathbf{S}$ which is computed based on the data. Thereby, the big advantage of spectral methods is the uniqueness of the obtained results such that, unlike non-metric MDS, for example, a reproducible visualization is obtained. The techniques deviate in the way in which the similarity matrix $\mathbf{S}$ is computed from the data.

As a simple example [98], CMDS can be seen as a spectral technique, since it transforms the matrix of pairwise Euclidean distances of data to the matrix of scalar products using double centring, and than referring to its eigenvalue decomposition. The projection takes the form $\mathbf{Y} = \mathbf{I}_{D' \times N} \mathbf{\Lambda}^{1/2} \mathbf{U}^\top$, where $\mathbf{Y}$ is the matrix of D'-dimensional projections $\mathbf{y}_i$ and $\mathbf{I}_{D' \times N}$ is the $N \times N$ unity matrix with D' ones on the diagonal.

Isomap is very similar to CMDS [147], the only difference being that, instead of Euclidean distances, geodesic distances of the original data manifold are used. The idea is, given a low-dimensional manifold embedded in high-dimensional space, to unfold the latter based on the distances computed along the manifold. The exact geodesic distances are usually not available, but an approximation is possible based on the data, assuming a locally almost flat structure. Then, the local neighbourhood graph reliably describes the local distance structure and global distances can be based thereon taking shortest distances in the graph. Typically, the graph is constructed by taking all points as vertices and adding edges between k-nearest neighbours or all neighbours in an ϵ-environment, respectively. The choice of k or ϵ is thereby an important parameter of the method. If it is chosen too large, the manifold can not be unfolded; if it is chosen too small, the produced graph may be not connected and only parts of the data can be visualized. Isomap yields excellent results for locally two-dimensional data sets, as can be seen in Figure 2.9. However, the behaviour for intrinsically higher dimensionality is unclear, as can be seen when inspecting the projection of the mnist data set as depicted in Figure 2.9. Clear class structures are hardly visible in this projection.

The preservation of the local neighbourhood structure of the data is also the goal of locally linear embedding (LLE) [132]. Here, the local linear relationships of data are preserved as much as possible. To achieve this, each point is represented as weighted linear combination of its k-nearest neighbours. This weighting should stay the same in high and low-dimensional space. The solution to this problem involves computation of an eigenvalue problem and the projections are given as the eigenvectors associated to the D' smallest eigenvalues unequal zero of this matrix. LLE is capable of unfolding the sphere, as can be seen in Figure 2.10, and to also display a few of the different clusters present in mnist, albeit not conveying the full structure in the latter case.

Similar to LLE, Laplacian eigenmaps (LE) aim at preserving the neighbourhood structure of the given data [8]. To achieve this, a neighbourhood graph is constructed

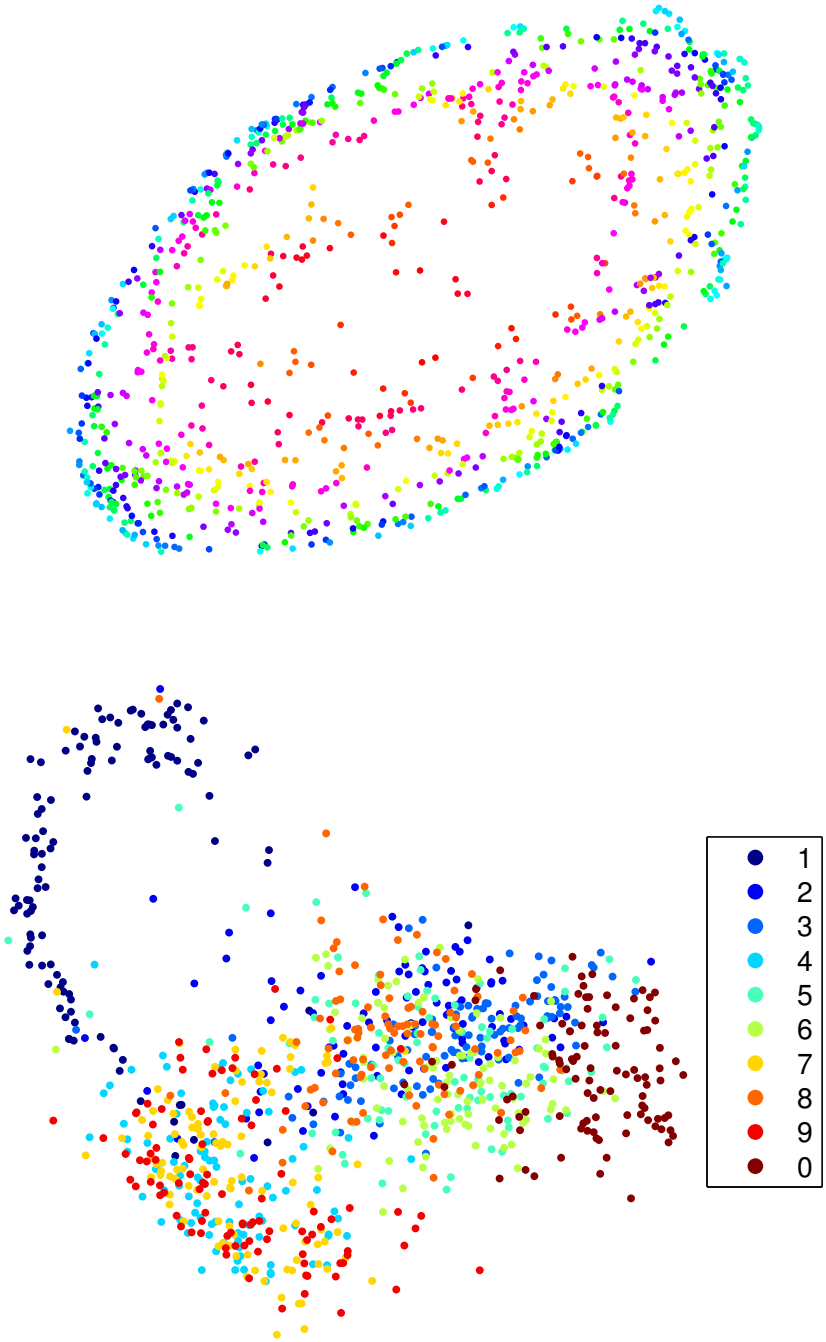


Figure 2.9: Results for Isomap for the two data sets **sphere** (top) and **mnist** (bottom). The color represents the position in the 3rd dimension for **sphere** and the class for **mnist**.

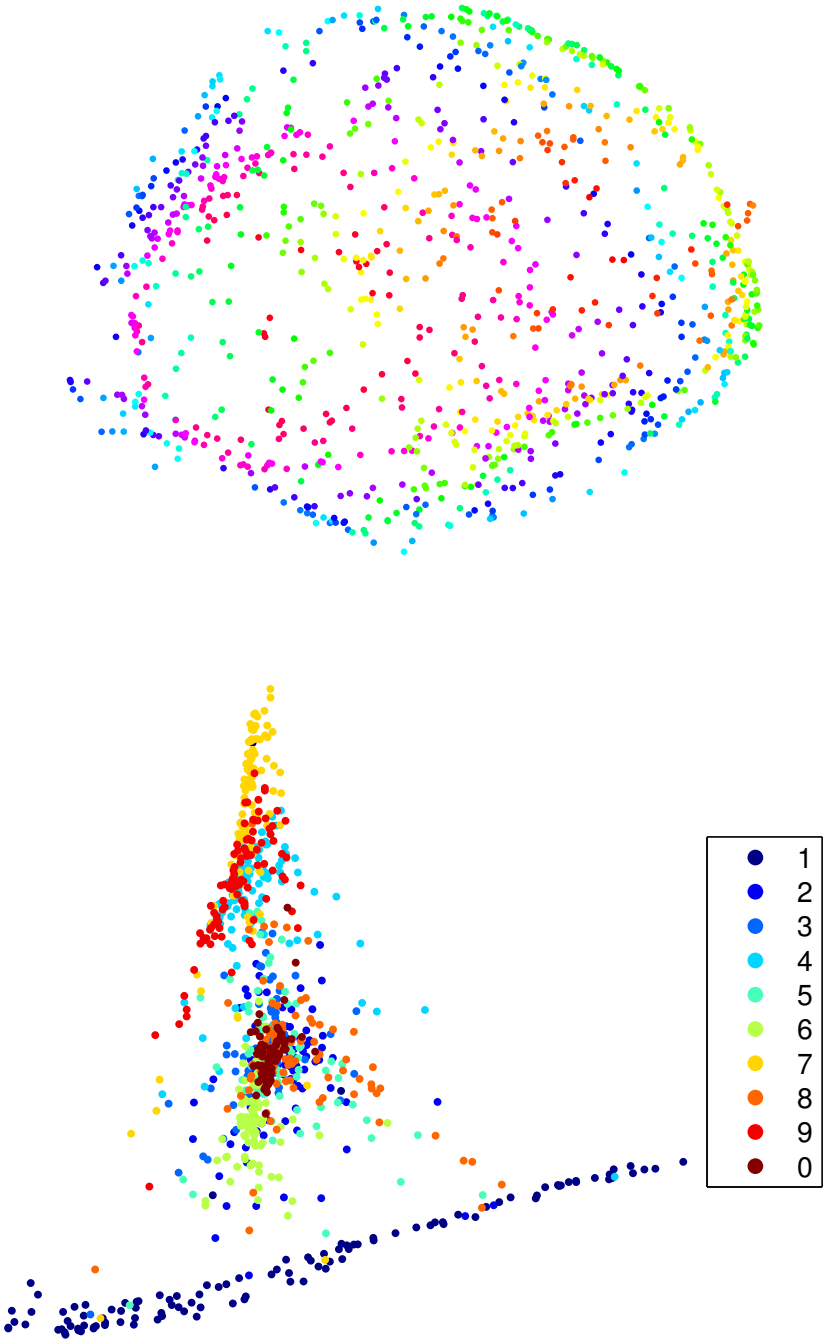


Figure 2.10: Results for locally linear embedding for the two data sets **sphere** (top) and **mnist** (bottom). The color represents the position in the 3rd dimension for **sphere** and the class for **mnist**.

which induces the corresponding graph Laplacian which depends on the distances of these local neighbours. Again, the projections are given as the eigenvectors of the graph Laplacian associated to the D' smallest eigenvalues unequal zero. As can be seen in Figure 2.11, the integration of the full graph Laplacian yields a clearer unfolding of the sphere, while still only a few cluster structures become apparent for the mnist data.

Maximum variance unfolding (MVU) preserves the distances of a data point to its closest k nearest neighbours as much as possible [170]. While these neighbourhoods are preserved, the variance between unconnected points is maximized, thus the manifold can be unfolded. The cost function is optimized by using semidefinite programming to compute a kernel matrix which fulfils the necessary constraints. The projection is then given by the eigenvectors associated to the D' largest eigenvalues.

Maximum entropy unfolding (MEU) is closely related to MVU and motivated by its background [92]. Again, this technique tries to preserve the local neighbourhood distances as far as possible. Unlike MVU, it defines a probability distributions of the data points and maximizes the entropy of it to obtain a uniform distribution of the data apart from the relations determined by the neighbourhood structure. The maximum likelihood solution of this probability gives a similarity matrix which can be visualized in a similar way as in CMDS. It has been shown in [92] that the alternatives Isomap, LLE, LE, and MVU are strongly connected to MEU and in fact can be interpreted as approximations of the projection given by MEU. In consequence, the projections of typical data manifolds look pretty similar, as can be seen in Figures 2.9 to 2.12.

Since all these techniques are nonparametric, they do not provide an explicit mapping of the data. In consequence, these methods are not capable of directly visualizing a new data point which has not been used during training, and extra effort is necessary to achieve this, as explained e.g. in [9, 18, 132]. Some of the techniques have been turned into parametric, mostly linear methods such as the linear mappings neighbourhood preserving embedding [70] and orthogonal neighbourhood preserving projections [89], and locally linear coordination (LLC) motivated by LLE [130], or the linear technique locality preserving projections [71] motivated by Laplacian eigenmaps. In addition to these variants, it is also possible to kernelize some of these approaches, yielding nonlinear extensions, or to integrate auxiliary information in the form of supervision, a principle, which we will detail in Section 2.4.2.

2.2.3 Neighbourhood preserving techniques

The nonparametric DR techniques discussed above aim at preserving pairwise distances while projecting the data. These distances might be original, weighted or nonlinearly preprocessed distances. Some techniques, such as e.g. LLE or LE, focus explicitly on small distances, in order to preserve the local structure of the data. The same motivation gives rise to a multitude of methods which focus on the local structure by preserving the neighbourhoods. This principle already guides SOM which due to its parametrization converges to locally linear approximations of the data space. It is possible to extend these ideas towards nonparametric approaches which are capable of following locally nonlinear structures as well. Like SOM, some of these techniques are based on heuristics rather than an explicit cost function, and all methods usually possess local optima such that

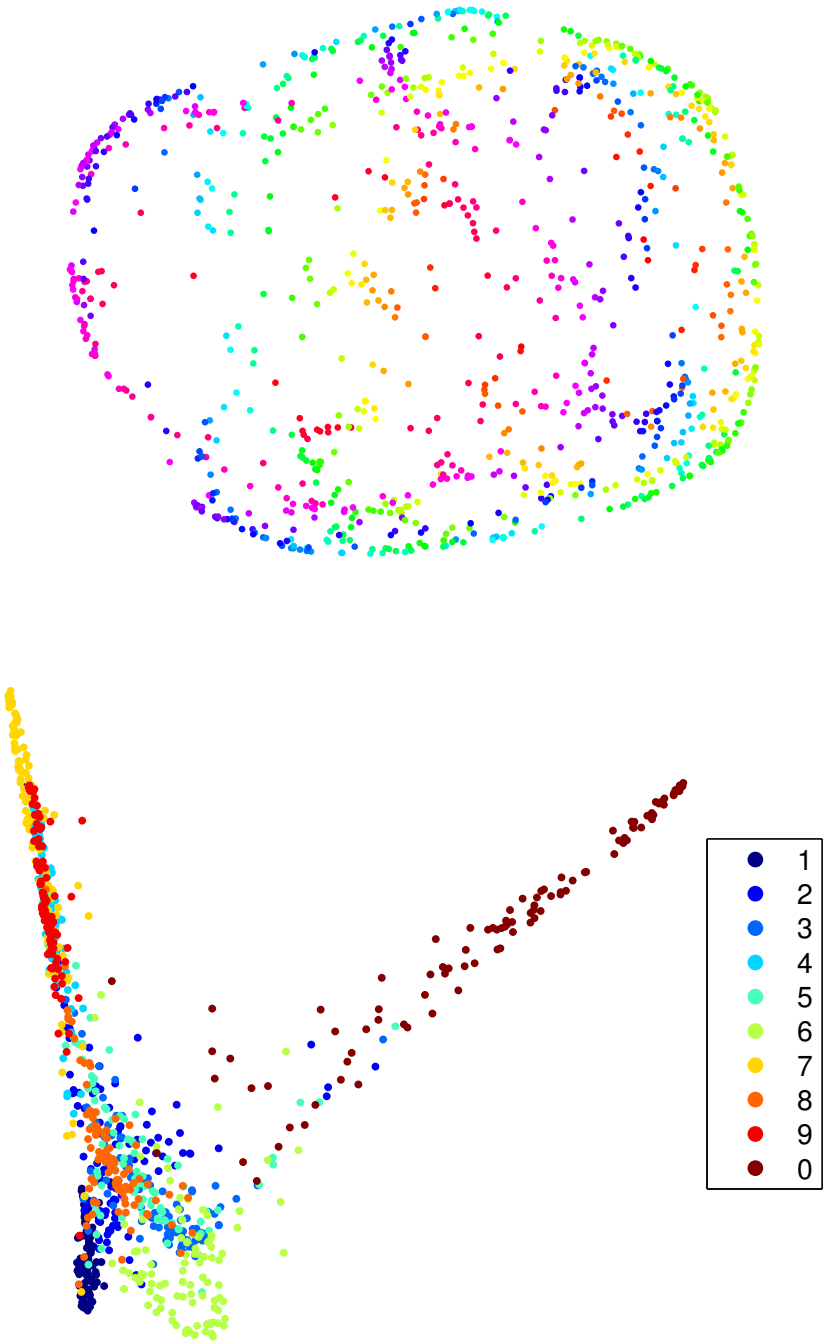


Figure 2.11: Results for Laplacian eigenmaps for the two data sets **sphere** (top) and **mnist** (bottom). The color represents the position in the 3rd dimension for **sphere** and the class for **mnist**.

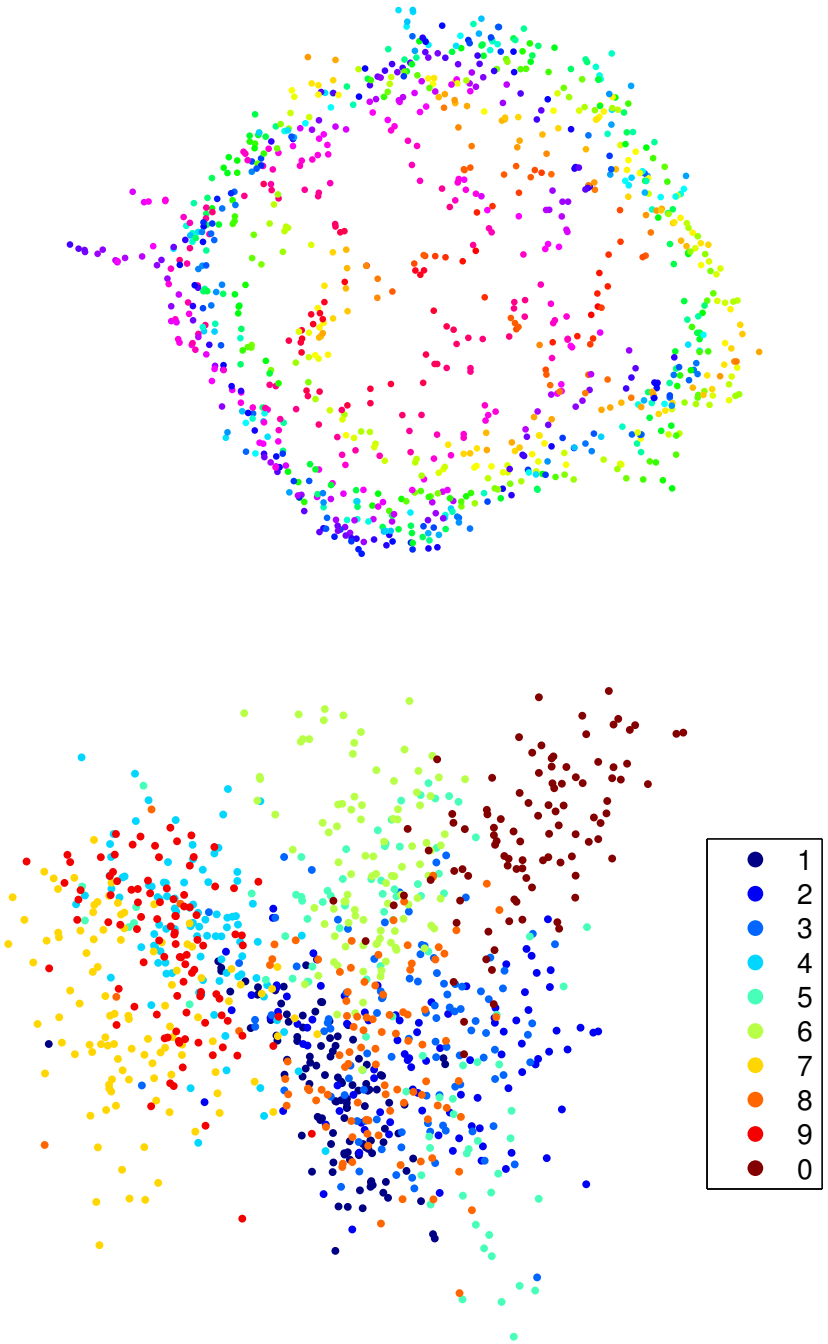


Figure 2.12: Results for maximum variance unfolding for the two data sets **sphere** (top) and **mnist** (bottom). The color represents the position in the 3rd dimension for **sphere** and the class for **mnist**.

results are non-deterministic.

One early method is given by Isotop [97] which tries to overcome the problems of SOM induced by the priorly fixed topology. Isotop, in contrast, builds a local neighbourhood graph directly on the data or a quantization of it, respectively. This data driven lattice is then unfolded in the plane by using an iterative update which attracts the lattice points towards data which are randomly generated on the plane driven by the current embedding of the lattice points. The exploration machine (XOM) is very similar to Isotop, the only difference being that it directly relies on pairwise distances in the original data space rather than a precomputed lattice given by the local neighbourhoods.

In contrast to these heuristic approaches, stochastic neighbour embedding (SNE) [74] relies on an explicit cost function for the preservation of pairwise distances of the data. Essentially, these distances are turned into probabilities of points being neighbours using a Gaussian model. Then, projections are determined such that the resulting distribution is as similar as possible to the original one as measured by the Kullback-Leibler divergence. One essential parameter of the method is the bandwidth of the Gaussians in the high-dimensional data space, which has to be adjusted locally to fit the local data density. Typically, an external global parameter, the so-called perplexity, is set, representing the number of effective neighbours as induced by the local bandwidths. The resulting cost function can possess local optima, and it is typically optimized by a gradient technique, such that the result is non-deterministic.

SNE suffers from a problem that is typically referred to as the crowding problem: in low dimensions, the space is rather limited as compared to the original high-dimensional data space, such that some distances necessarily have to be stretched, unless several regions of the space are mapped on top of each other. Since the cost function does not account for this effect, typically, points are clumped together by the resulting mapping. To overcome this problem t-distributed SNE (t-SNE) has been introduced in [158]. Instead of Gaussian probabilities in the projection space, the Student t-distribution is used, which is a long-tail distribution. Hence large distances can be matched by a wide range of scales in the projection, this way avoiding the crowding problem. t-SNE is particularly good in displaying cluster structures, as can be seen in Figure 2.13: different clusters of the mnist data set are clearly laid out on the visualization. It has been shown in [96] that a comparable effect can be achieved with SNE as well, if instead of a simple Kullback-Leibler divergence symmetric variants or divergences of higher order are used. One example of a symmetric difference is introduced in the neighbourhood retrieval visualizer (NeRV) which also has a counterpart in an information theoretic approach to DR [163]. t-SNE as one of the most prominent nonlinear nonparametric DR techniques available today, will be in the focus in chapter 6. There, we will address the question how to enhance t-SNE to obtain a parametric technique for easy out-of-sample extension and integration of prior knowledge.

The collection of algorithms described in sections 2.1 and 2.2 covers some of the most popular DR techniques available today, their properties, and their motivation. As demonstrated, the results of the techniques and their properties are rather diverse, such that different algorithms can give optimum performance in different settings. As a summary, the most relevant algorithms and their properties are shown in Table 2.1. As explained before, we distinguish the important properties of the techniques (i) being

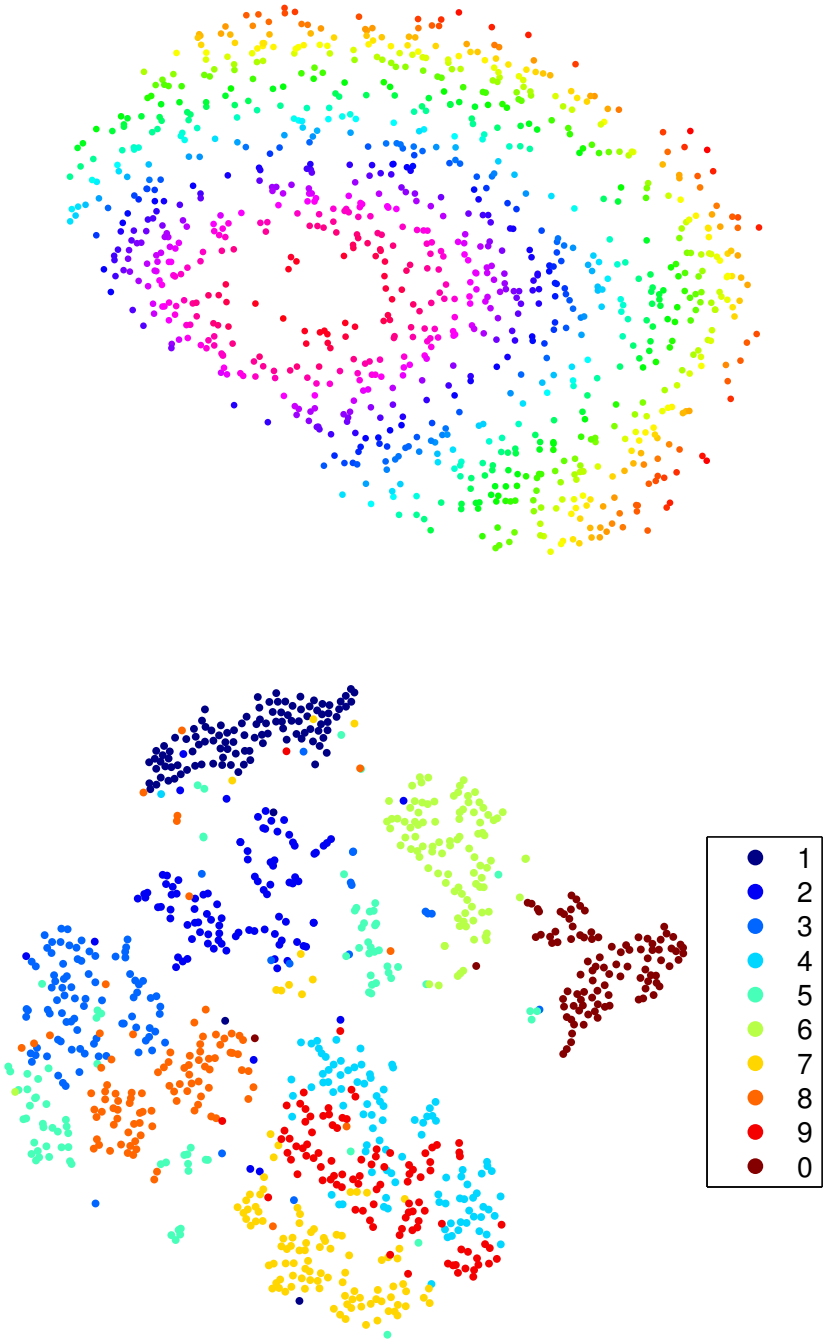


Figure 2.13: Results for t-distributed stochastic neighbour embedding for the two data sets **sphere** (top) and **mnist** (bottom). The color represents the position in the 3rd dimension for **sphere** and the class for **mnist**.

parametric or nonparametric, (ii) providing linear or nonlinear visualization, (iii) the data type the techniques are capable of dealing with, in some cases restricting to vectors, in others relying on a similarity or dissimilarity matrix only, (iv) the question whether the results are unique, or e.g. due to local optima different outcomes can be present in different runs of the algorithm, (v) the underlying objective, (vi) and whether the latter is formulated in terms of a mathematical cost function or realized via heuristic terms. There exists a variety of further techniques which we do not mention here such as extensions of MEU based on its modelling via Gaussian random fields [92], extensions of XOM or t-SNE to more general divergences [19], alternative kernel approaches to DR [145], alternative probabilistic models [109], or further spectral techniques for DR [31].

2.3 Evaluation

Due to a large amount of DR techniques with different motivations and approaches to the problem, naturally, the question arises how to compare the methods and how to objectively evaluate the performance of the techniques for a given data set. Eventually, the suitability of a DR technique depends on the concrete setting since there is no general objective for the unsupervised problem of DR. Nevertheless, a few principled evaluation measures have been proposed which quantitatively measure in how far the results conform with a specific objective of DR.

Note that the development of the evaluation techniques is parallel to the development of the visualization techniques and any DR technique which is based on a mathematical cost function also provides a quantitative evaluation measure and vice versa. Still, evaluation techniques can help to analyse which properties an algorithm has and which algorithm is the most suitable to the problem at hand. Furthermore, quality evaluation can be used to detect convergence issues and to handle the automatic parameter selection of specific algorithms.

One principled approach is based on the notion of information preservation in terms of the reconstruction error of the data. Assume not only a projection $\mathbf{x}_i \to \mathbf{y}_i$ of data is available but there exists also a way to judge in how far the original data point can be reconstructed given the projection only $\mathbf{y}_i \mapsto \tilde{\mathbf{x}}_i$. This inverse projection is explicitly given for auto-encoder networks, for example. For techniques such as PCA or SOM, the reconstruction mapping is provided by the pseudo-inverse or the weights attached to a lattice point, respectively. In these cases, one can measure the reconstruction error in a mean squared error sense (MSE): $\mathrm{MSE} = 1/N \sum_{i=1}^{N} \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|^2$. This cost function is directly optimized by PCA or auto-encoder networks. In particular for nonparametric approaches, however, the notion of an inverse mapping $\mathbf{y}_i \mapsto \tilde{\mathbf{x}}_i$ is not defined.

Often auxiliary information is available for the given data in form of class label information. In such cases, one can measure in how far a projection respects this auxiliary information by evaluating in how far classes separate in the projection. A simple way to measure this property is offered by the k-nearest neighbour (kNN) technique. For fixed k, the label of a given point is compared to the majority label of its k nearest neighbours. If the amount of misclassifications is increased significantly by the DR as opposed to the original data, this can act as an indicator that the projection is not suited under

Table 2.1: Some popular dimensionality reduction techniques and their characteristics

method	type	effort	data type	deterministic	objective	cost function
PCA	linear parametric	$\mathcal{O}(N)$	vectors	yes	variance preservation	yes
Original SOM	nonlinear parametric	$\mathcal{O}(N)$	vectors	no	topology extraction	no
GTM	nonlinear parametric	$\mathcal{O}(N)$	vectors	no[a]	manifold generation	yes
Autoencoder	nonlinear parametric	$\mathcal{O}(N)$	vectors	no	information preservation	yes
MDS	nonlinear nonparametric	$\mathcal{O}(N^2)$	distances	yes	distance preservation	yes
Isomap	nonlinear nonparametric	$\mathcal{O}(N^3)$	distances	yes	manifold extraction	yes
LLE	nonlinear nonparametric	$\mathcal{O}(N^2)$	vectors	yes	manifold extraction	yes
LE	nonlinear nonparametric	$\mathcal{O}(N^2)$	distances	yes	manifold extraction	yes
MVU	nonlinear nonparametric	$\mathcal{O}(N^2)$	distances	yes	manifold extraction	yes
MEU	nonlinear nonparametric	$\mathcal{O}(N^2)$	distances	yes	manifold extraction	yes
Isotop	nonlinear nonparametric	$\mathcal{O}(N^3)$	distances	no	topology extraction	no
XOM	nonlinear nonparametric	$\mathcal{O}(N^2)$	distances	no	topology extraction	no
SNE	nonlinear nonparametric	$\mathcal{O}(N^2)$	distances	no	neighbourhood preservation	yes
t-SNE	nonlinear nonparametric	$\mathcal{O}(N^2)$	distances	no	neighbourhood preservation	yes
LDA	linear parametric	$\mathcal{O}(N)$	vectors	yes	classification performance	yes
Rel. SOM, GTM	nonlinear parametric	$\mathcal{O}(N^2)$	distances	no	topology preservation	yes
LiRaM LVQ	nonlinear parametric	$\mathcal{O}(N)$	vectors	no	classification performance	yes
LMNN	linear parametric	$\mathcal{O}(N^2)$	vectors	yes	kNN error minimization	yes

[a] if initialized with PCA directions, determinism can be achieved, albeit possibly only local optima of the costs are achieved

the assumption that the given auxiliary information contains relevant information of the data. Interestingly, a k nearest neighbour labelling approach has also been taken in [159] to evaluate the preservation of local neighbourhood structures; in this case labels are attached to the original data by a simple uniform tessellation of the original space. It should be noted though, that this approach is only meaningful if the labelling has a reasonable structure, i.e. the labelling is smooth as regards the data topology and classes are balanced. After all, the goal of DR is not to achieve the best possible classification accuracy, but instead to find a low-dimensional projection which correlates with the structure of the original data as far as possible. Because of this observation, this form of evaluation measure has been used together with a manually created labelling of the data which respects the topological ordering [159]. Data in the original feature space are decomposed into two classes by means of a regular checkerboard structure for example, resulting in balanced classes which respect the data topology. Any decrease in accuracy would indicate structural mistakes made by the evaluated DR technique.

In the last years, evaluation measures have often focussed on the structure preservation properties of the DR techniques, this way accounting for the intuition that the exact scaling of the data becomes irrelevant while projecting, but the overall data shape should be preserved. Early approaches to quantify structure preservation have already been developed in the context of self-organizing maps: To measure if a SOM is unfolded correctly with respect to the data the topographic product has been proposed [7] and later modified to the topographic function [164], which can better take into account curvatures of the data manifold. Essentially, the topographic product tests for each neuron whether its neighbours in the data space are the same as in the SOM grid. Depending on this information it is possible to compute whether the network lies on a smooth manifold. If the topographic product is negative, then the dimensionality of the network is lower than the intrinsic dimensionality of the data and the network has to fold itself to cover the data space. On the other side, if the topographic product is positive, then the dimensionality of the network is higher than that of the data. The topographic function $\Phi(k)$ gives more insight into the topology preservation and is defined on the interval $[-K, K]$, with K being the number of neurons. A mapping is topology preserving iff $\Phi \equiv 0$ and especially iff $\Phi(0) = 0$, since $\Phi(0)$ is an agglomeration of Φ. The values of k for which $\Phi(k) \neq 0$ show where the topology is disturbed. If $\Phi(k)$ deviates from zero for small values of $|k|$, then the disturbance is present on the local scale. For a large value of $|k|$ flaws on the global scale are indicated. If the dimensionality of the network is lower than the intrinsic dimensionality of the data, then $\Phi(k) \neq 0$ only for $k > 0$. On the other side $\Phi(k) \neq 0$ for all values of k if the dimensionality of the mapping is too big. Although the topographic product and the topographic function give insight into the topology and neighbourhood preservation, their computation is quite costly, and they have been proposed for techniques which use prototypes on a regular grid only.

Trustworthiness and continuity [83] can be seen as a transfer of these ideas to general nonparametric mappings. The measures address the neighbourhood preservation based only on the original and the projected data, independent of the used mapping technique. Essentially, trustworthiness and continuity count how many neighbours of the original data are preserved while projecting and vice versa, respectively. A trustworthy visualization guarantees that points shown next to each other in the projection are also neighbours

in the high-dimensional space. On the other hand, continuity measures if there are any tears in the visualization, by counting how many points extrude the neighbourhood of the original data space. These two quantities are evaluated for all neighbourhood sizes, thus giving functions which show the quality of DR on different scales. A similar technique called local continuity meta-criterion has been proposed by [23]. It measures the overlap between the neighbourhoods of size k in high- and low-dimensional spaces. Similarly, the mean relative rank error [98] measures rank deviations in neighbourhoods relative to the rank, since for already high ranked neighbours small permutations are not critical.

In [99] a unifying framework for rank based criteria has been proposed. The above criteria can be derived thereof by a summation over different blocks of the so called co-ranking matrix. This $(N-1) \times (N-1)$ matrix Q with elements q_{kl} counts for how many points the rank to some other point changed from k to l while projecting the point. Interestingly, the entries of the matrix obey certain invariances. Because of this insight, a quality measure has been proposed in [99] which describes the overall quality of a mapping and whether the mapping has intrusive or extrusive behaviour [100]. The same authors also propose scale-independent quality criteria, which characterize the quality function which is a graph depending on the neighbourhood size by only two scalar values characterizing their local and global behaviour[101].

It has been proposed in [113] to use this quality measure as a point-wise indicator about the reliability of a given mapping and to explicitly integrate this information into the visualization by means of color coding. This way, the user can get an intuition about which parts of the visualization might be erroneously displayed. Since an intuitive interpretability of the displayed colors is crucial for this procedure, it has been proposed in [113] to take the sums over a slightly different part of the co-ranking matrix in a much more intuitive way such that the parameters involved in the summation have a natural interpretation for a human observer: the interesting neighbourhood size and the tolerated error range, respectively. This formulation allows to analyse different kinds of errors in more detail, e.g. the preservation of small neighbourhoods, global relationships of the data, or detection of tears.

2.4 Recent developments

The topic of nonlinear DR is subject of quite some research regarding diverse topics such as e.g. its combination with classical information visualization approaches, its extension towards dynamic data, mathematical foundations and guarantees of the algorithms, or challenging applications. Here, we exemplarily highlight three topics in which some results could be achieved recently.

2.4.1 General data structures

DR in its original form addresses the projection of high-dimensional vectors to a low-dimensional space. Often, however, data are not given in form of vectors, but as pairwise relations between data points, or data possess additional structural elements such as a time dynamics, or an underlying graph structure.

There have been quite some efforts to develop DR techniques for structures such as graph structures or time series, see e.g. [139] for a very promising graph drawing approach developed in the context of machine learning, or the overviews [5, 63] for extensions of the self-organizing map to deal with temporal or structural data by means of recursive processing. In this overview, we shortly have a glimpse at a more general approach, which treats arbitrary data in terms of pairwise similarities or dissimilarities only. This opens the way towards quite general structures, since a variety of dedicated metrics exist in this context such as alignment distances for biological sequence data, time warping for times series objects, or the normalized compression distance to compare texts [61].

Note that many DR techniques are based on the notion of distances or more general dissimilarities, such that they can directly be applied to such cases; this includes, for example, distance based techniques such as MDS, t-SNE, or XOM as well as methods which built on a neighbourhood graph such as Isomap or CCA. Prototype based techniques on the other hand, such as SOM or GTM, need a vector space to represent positions of the prototypes. As suggested in [69], it is possible to express the prototypes as an implicit linear combination of the data points even though the explicit positions of the points are not known. The resulting algorithm relational SOM [60] produces the same results as SOM, if the distances are computed from Euclidean vectors. In the case that the distances are not metric, the algorithms can be related to an implicit embedding of data in the so-called pseudo-Euclidean space [59, 119]. The extension of GTM to a relational counterpart will be the subject of chapter 4.

2.4.2 Discriminative dimensionality reduction

Since DR is an ill posed problem, many different paradigms of which information to preserve have been developed. These formalizations, however, have the drawback that they are based on mathematical terms rather than the intuition of the user, thus allowing the user little possibility to explicitly shape the visualization according to the aspects he/she thinks most important.

One very promising way to allow a human observer to influence the results of the DR method is framed under the umbrella of discriminative DR: here, data are provided together with auxiliary information, such as class labels for each data point. This information is given by the user to shape the method in such a way that the information which is most relevant for the given class labelling should be visualized.

A variety of different techniques has been proposed to incorporate auxiliary information in the form of class labels into a visualization technique, see e.g. [30, 43, 141] for ad hoc modifications of existing approaches or [18, 21, 84, 125] for diverse principled approaches. One of the most popular discriminative visualizers is offered by linear discriminant analysis (LDA) [39], a parametric linear technique very similar to PCA. Here, the projection matrix $\mathbf{L}$ is chosen in such a way, that the distance between different classes is maximized and the distance inside each class is minimized. In simple cases LDA can produce excellent results, however, it cannot cope with nonlinearities in the data or multimodal class structures. To partially overcome these problems kernelized approaches such as general discriminant analysis [6] and kernel clustering-based discriminant analysis [106] have been proposed.

Another linear technique motivated by PCA is the Neighbourhood Components Analysis (NCA) [51]. Instead of using the Euclidean metric in the data space, it learns a quadratic metric in such a way, that k-nearest neighbour performance is maximized. If the learned metric is chosen to be of a low rank, then NCA can be used for DR. Large margin nearest neighbour (LMNN) [171] pursuits this idea even further. It reformulates the problem in terms of semidefinite programming, so that it becomes convex and efficiently solvable. The linearity of the technique, however, limits its flexibility as can be seen in Figure 2.14.

A general learning metrics principle has been proposed for SOM [84, 125]. Here, a flexible Riemannian metric is defined which is induced by the Fisher information matrix at a given data point. The local tensor is thereby learned based on the training data with labels in such a way, that the distance between neighbours with the same class is shortened and enlarged for different classes. Here, direct nonparametric methods or, alternatively, more efficient parametric approaches can be used to capture the label information. Since the computation of minimum distances in terms of Riemannian path integrals is quite costly, several approximations have been proposed [84, 125]. This way, the learning metrics principle allows to compute discriminative distances between points in a principled way which can then be plugged into any distance based visualizer such as t-SNE or NeRV [124], see Figure 2.16 for an example visualization for the mnist data. This approach yields very good results but at the cost of a high computational load. We will discuss the integration of this learning metrics principle into parametric kernel t-SNE in chapter 6.

To overcome this problem limited rank matrix learning vector quantization (LiRaM LVQ) has been proposed [20]. The idea is to approximate the Riemannian metric by a locally constant quadratic metric. This is achieved by clustering the data using vector quantization into areas with locally similar metric. Each cluster can be represented by a prototype with attached quadratic metric. Similar to NCA the rank of the metric can be limited to produce a linear DR mapping. The areas represented by prototypes can be glued together using manifold charting to obtain a globally nonlinear mapping. A similar approach can be applied to GTM [44], as we will show in chapter 3.

An alternative approach to perform DR tuned by human observer, without auxiliary information, was presented in [127]. The idea is to allow the user to modify parameters of a DR technique and to update the low-dimensional projection in real time. Thus, allowing to discover different aspects of the data in an interactive way. In order to be responsive, this approach precomputes a large number of low-dimensional projections, one for each possible combination of parameters sampled on a grid, and interpolating the projections in between. This, of course, requires a large amount of computational time and it is desirable to develop fast techniques, which might be even able to compute the updated projections on the fly. We will address this topic in the next section.

2.4.3 Scalability

Due to modern technology, which results in more precise and faster sensors, it is possible to collect not only very high-dimensional data but also large amounts of data points. The issue of big data sets constitutes one of the major challenges of information technology

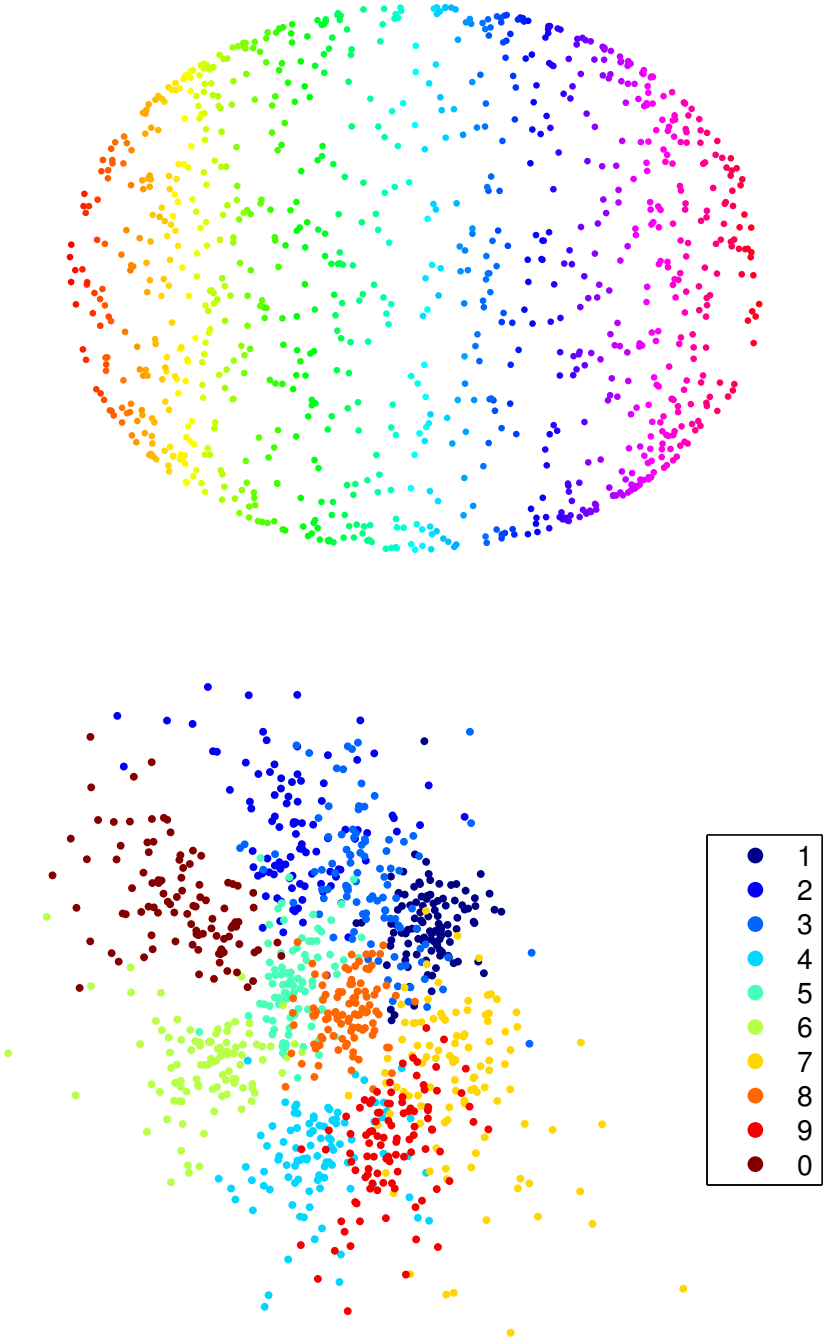


Figure 2.14: Results for large margin nearest neighbour for the two data sets **sphere** (top) and **mnist** (bottom). The color represents the position in the 3rd dimension for **sphere** and the class for **mnist**.

these days [148]. Often, nonlinear DR techniques take into account pairwise relations between data points, such that their computational effort is at least quadratic in the number of points, hence methods are infeasible already for medium sized data sets. Techniques relying on a complete eigenvalue decomposition, e.g. spectral methods, often have even $\mathcal{O}(N^3)$ runtime complexity. In order to overcome this problem different approaches have been taken.

Heuristic approaches and approximations

The goal of the High-Throughput MDS (HiT-MDS) [143] is to optimize the Pearson correlation coefficient between the high and low-dimensional distances. First the dimensionality of the data is reduced by a random projection to generate a sufficient initial solution. Then the cost function is optimized iteratively taking into account the gradient regarding only one point at a time. The algorithm converges when for all points no further improvements can be made. Since each step requires only linear amount of memory and time, this algorithm is well suited for online applications for big data sets.

Recently, a very powerful speed-up technology has been proposed for neighbour embedding techniques such as t-SNE [157, 174]. The underlying ideas stem from approximations used in physics to simulate multiple particle systems. For the simulation to be trustworthy the forces between all pairs of particles have to be considered or, due to their number, efficiently approximated. The Barnes-Hut algorithm [4] creates a tree which hierarchically separates the data into groups. Depending on the distance to a point, a group can be approximated by its mean, thus reducing the number of computations significantly. The approximation allows to reduce the complexity of embedding techniques such as t-SNE or neighbour embedding to $\mathcal{O}(N \log(N))$.

Nyström approximation technique

One possibility to reduce the complexity of DR approaches which are based on a similarity or dissimilarity matrix is to use the Nyström approximation of the matrix. This approximation technique has been proposed in the physical domain by [117] and later reintroduced to machine learning by [173] to approximate positive semi-definite matrices in the context of the support vector machine. Recently it has been shown [49] that it can also be applied to an arbitrary symmetrical matrix.

To apply the Nyström approximation to N points, one first selects a small subset of m points, so called landmarks. Typically they are chosen randomly, but there exist more sophisticated selection techniques [177]. A (dis-)similarity matrix $\mathbf{K}$ can then be approximated by taking into account only a linear part of this matrix. The approximation has the form $\mathbf{K} \approx \mathbf{K}_{N,m}\mathbf{K}_{m,m}^{-1}\mathbf{K}_{N,m}^{\top}$ where $\mathbf{K}_{N,m}$ consists of the (dis-)similarities between all N points and m landmarks; $\mathbf{K}_{m,m}^{-1}$ is the Moore-Penrose pseudo inverse of the matrix containing the (dis-)similarities between the landmarks. This way, for a small m, only a linear part of the matrix needs to be computed and stored. Further, using this decomposition in matrix based learning algorithms, yields to linear time computation only if matrix vector operations are evaluated in a suitable order. Some algorithms can be modified accordingly to employ this technique and reduce the memory complexity to

$\mathcal{O}(mN)$ and the computational complexity to $\mathcal{O}(m^2N)$.

Obviously, the Nyström approximation is suitable for spectral techniques and it has been applied e.g. in Landmark MDS and landmark Isomap [33]. In [9] this idea has been extended towards a general approach covering MDS, Isomap, LLE, and LE. Also, the Nyström method can be applied for prototype based techniques which operate on (dis-)similarities as has been shown for relational GTM [49]. In chapter 5, we will discuss possibilities to obtain linear time methods for techniques based on similarities or dissimilarities in a general way.

It has been stated in [24] that an approximated solution does no longer optimize the desired cost function and even does not fulfil the stated constraints, if any. Correspondingly, the Nyström technique for MVU has been extended to so-called Maximum Variance Correction [24], which corrects the errors produced by the Nyström approximated manifold learning techniques.

Explicit mapping

One general approach to large data sets relies on the assumption that already a subset of all data carries the relevant information to learn the principled shape of the mapping, all other data are not necessary to infer the overall structure. Based on this assumption, it has been proposed in [18, 46] to infer a DR based on a subsample of all points only, obtaining the visualization of all data afterwards. For this principle, however, it is mandatory to obtain an explicit mapping function hence powerful nonparametric techniques cannot directly be used for this principle.

Correspondingly, there has been some work how to infer a nonlinear function form the data rather than the projection points only, see e.g. [18, 46, 156]. In [156] an explicit function is given in the form of a deep neural network which is trained using a Boltzmann machine approach. This has the drawback of quite high computational load and, due to its large flexibility, a large number of necessary training data. In [18] a general approach is demonstrated using a linear function as well as locally linear functions in combination with the t-SNE costs, yielding respectable results. In chapter 6, we will propose a kernel based mapping:

$$f(\mathbf{x}) = \sum_i \alpha_i \cdot \frac{k(\mathbf{x}, \mathbf{x}_i)}{\sum_l k(\mathbf{x}, \mathbf{x}_l)}$$

where $\mathbf{x}_i$ are the training points, α_i mapping parameters which have to be trained, $k(.,.)$ a kernel function such as e.g. Gaussian kernel. Then, training reduces to a simple matrix inversion $\mathbf{A} = \mathbf{K}^{-1} \cdot \mathbf{Y}$ where the matrices $\mathbf{A}$, $\mathbf{K}$ and $\mathbf{Y}$ consist of α_i, $k(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{y}_i$, respectively. Since the mapping is trained on a small number of points m, it is fast, although its complexity is $\mathcal{O}(m^3)$. It is also capable to trustworthily infer the local data structure, as can be seen on the Figure 2.15.

A disadvantage of the approach is that, given only a small amount of points, some DR techniques such as t-SNE are partially not yet capable of learning the full structure underlying in the data. In such cases, additional knowledge in form of auxiliary label information might aid the visualization techniques. In [18] metric learning is used to achieve better clustering and thus more homogeneous areas for locally linear mapping. As

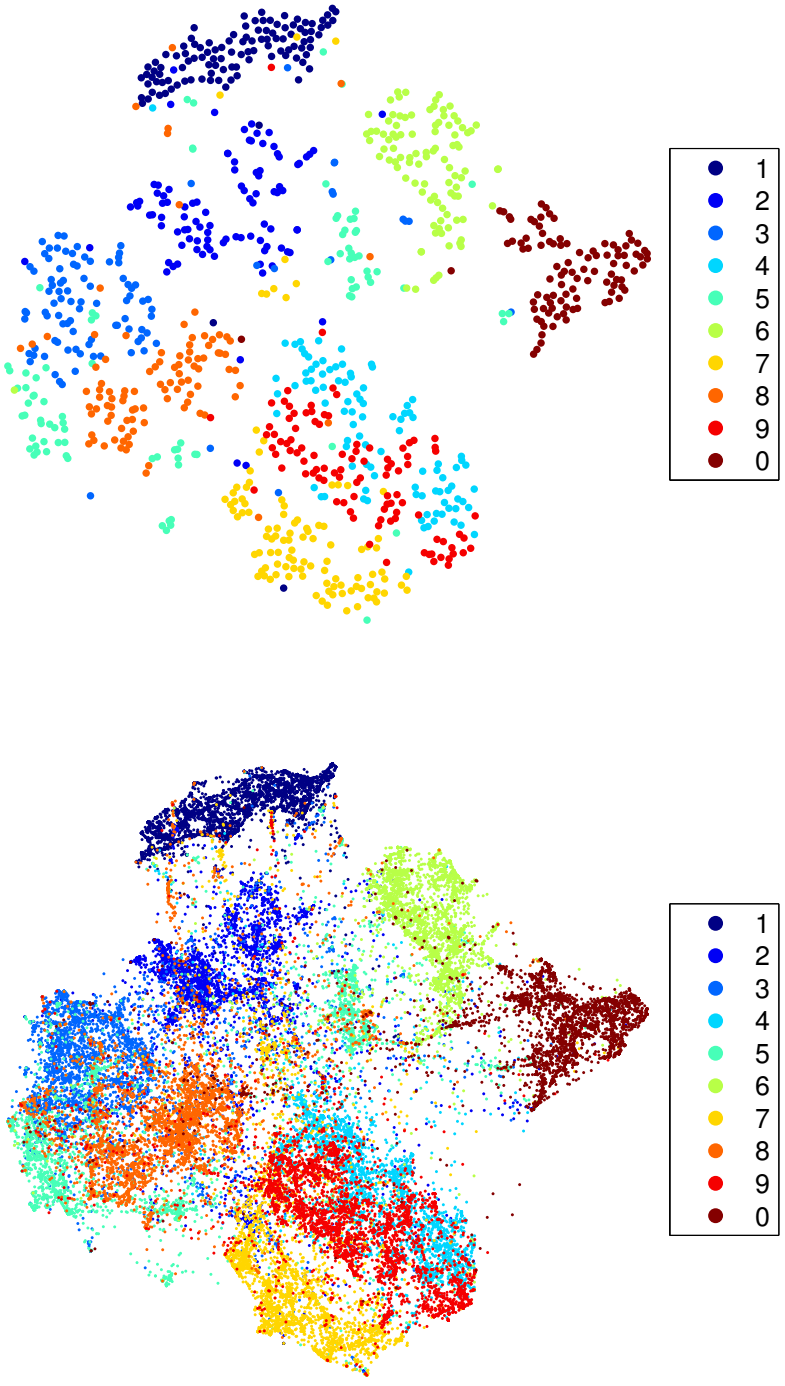


Figure 2.15: Results for t-SNE for the **mnist** data set (top), as well as its extension towards the full data set of about 60,000 data points (bottom).

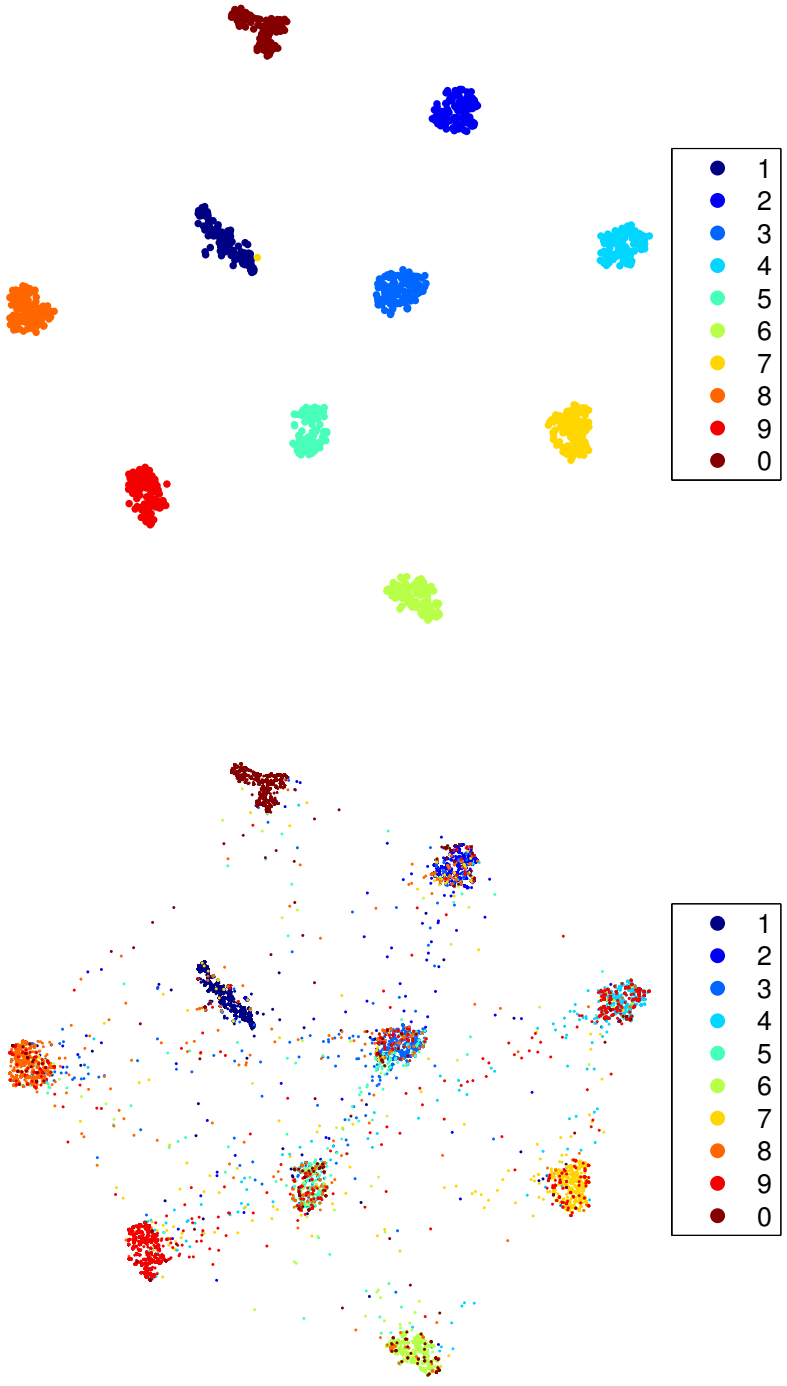


Figure 2.16: Results for Fisher-t-SNE for the **mnist** data set (top), as well as its extension towards the full data set of about 60,000 data points (bottom).

we will discuss in chapter 6, distance based techniques can easily be turned into supervised techniques by substituting Euclidean distance by the Fisher distance. The disadvantage of time consuming computations can be circumvented by combining Fisher distances with the kernel mapping, resulting in Fisher kernel t-SNE, which we will introduce in chapter 6. An example visualization of t-SNE enhanced by the Fisher information provided by the clusters as well as its generalization towards the full data set using a kernel mapping is displayed in Figure 2.16.

Although these techniques are presented only for kernel t-SNE in this overview, an extension of the ideas towards other nonparametric methods is obviously possible.

2.5 Summary

In this chapter, we have revised some of the most popular DR techniques used today for visualizing high-dimensional data in the plane and we have laid the ground for challenges, which will be tackled in the following chapter of this thesis. This way, along the structuring elements of parametric versus nonparametric techniques, we have covered very different mathematical formalizations to address the underlying problem of information preservation while projecting. This heterogeneity is also mirrored by a large diversity of evaluation measures, with general approaches such as the co-ranking framework just popping up in the last years.

Interestingly, as already indicated in a few examples, major problems are widely shared by different techniques. This fact gives rise to new paradigms which focus on solving these problems by adapting the existing techniques accordingly. One of these problems is the ability to deal with complex structured, relational data. Typically, it is not an issue for nonparametric techniques, since they often require only the distances between objects, but is problematic for parametric techniques, as they require a vector space to define a projection mapping. On the other hand, scalability, which is characterized by efficient out-of-sample extension and low computational complexity, is a positive characteristic of parametric techniques, while nonparametric techniques suffer from being slow and not having an explicit mapping. Finally, we have relevance learning, which redefines the goal of DR by focusing on preservation of structures according to auxiliary information, thus making the problem less ill-posed.

Table 2.2: Questions treated in this thesis.

Topic	Relational	Out-of-Sample	Efficiency		Relevance
Technique	Data	Extension	vectorial	relational	Learning
GTM	chapter 4	$\surd$	$\surd$	chapter 5	chapter 3
t-SNE	$\surd$	chapter 6	chapter 6		chapter 6

In the following chapters, as noted in the Table 2.2, we will address these problems in more detail on the example of two techniques, parametric GTM and nonparametric t-SNE.

This chapter is based on: Andrej Gisbrecht and Barbara Hammer. Relevance learning in generative topographic mapping. *Neurocomputing*, 74(9):1351–1358, 2011.

Chapter 3

Relevance learning in generative topographic mapping

Generative topographic mapping (GTM) has been introduced as a generative statistical model corresponding to the classical self-organizing map for unsupervised data inspection and topographic mapping [12]. An explicit statistical model has the benefit of great flexibility and easy adaptability to complex situations by means of appropriate statistical assumptions. Further, by offering an explicit mapping of the latent space to the observation space and a constrained Gaussian mixture model based thereof, GTM offers diverse functionality including visualization, clustering, topographic mapping, and various forms of data inspection. Like standard unsupervised machine learning and data inspection methods, however, GTM shares the 'garbage in - garbage out' problem: the information inherent in the data is displayed independent of the specific user intention. Hence, if 'garbage' is present in the data, the structure of this 'garbage' is presented to the user since the statistical model has no way to identify the information important to the user.

In this chapter, we extend GTM to the principle of learning metrics by combining the technique of relevance learning as introduced in supervised prototype-based classification schemes and the prototype-based unsupervised representation of data as provided by GTM. We propose two different ways to adapt the relevance terms which rely on different cost functions connected to prototype-based classification of data. Unlike [17], where a separate supervised model is trained to arrive at appropriate metrics for unsupervised data visualization, we can directly integrate the metric adaptation step into GTM due to the prototype-based nature of GTM. We test the ability of the model to visualize and cluster given data sets on a couple of benchmarks. It turns out that, this way, an efficient and flexible discriminative data mining and visualization technique arises.

3.1 Related work

The domain of data visualization by means of DR techniques constitutes a matured field of research, many powerful nonlinear reduction techniques as well as a Matlab implementation being readily available, see e.g. [8, 53, 98, 132, 147, 158, 159, 168, 169]. However, researchers in the community start to appreciate that the inherently ill-posed problem of unsupervised data visualization and DR has to be shaped according to the

user's needs to arrive at optimum results. This is particularly pronounced for real-life data sets which frequently do not allow a widely loss-free embedding into low dimensionality. Therefore, it has to be specified which parts of the available information should be preserved while embedding.

On the one hand, formal evaluation measures have been developed which allow an explicit formulation and evaluation based on the desired result, see e.g. [100, 161, 162, 163]. On the other hand, researchers start to develop methods which can take auxiliary information into account. This way, the user can specify which information in the data is interesting for the current situation at hand by means of e.g. labelled data.

There exist a few classical mechanisms which take class labelling into account to reduce the data dimensionality: Feature selection constitutes one specific type of DR. Feature selection constitutes a well investigated research topic with numerous proposals based on general principles such as information theory or dedicated approaches developed for specific classifiers; see e.g.[56] for an overview. However, this way, the DR is restricted to very simple projections to coordinate axes.

Several classical discriminative DR tools apply more flexible, but still linear projection methods: Fisher's linear discriminant analysis (LDA) projects data such that within class distances are minimized while between class distances are maximized. One important restriction of LDA is given by the fact that, this way, a meaningful projection to dimensionality at most $c-1$, c being the number of classes, can be obtained. Hence, for two class problems only a linear visualization is found. Partial least squares regression (PLS) constitutes another classical method which objective is to maximize the covariance of the projected data and the given auxiliary information. It is also suited for situations where data dimensionality is larger than the number of data points; in such cases a linear projection is often sufficient and the problem is to find good regularizations to adjust the parameters accordingly. Informed projection [30] extends principal component analysis (PCA) to also minimize the sum squared error of data projections and the mean value of given classes, this way achieving a compromise of DR and clustering in the projection space. Another technique relies on metric learning according to auxiliary class information. For a metric which corresponds to a global linear matrix transform to low dimensionality this results in a linear discriminative projection of data, as proposed e.g. in [20, 51].

Modern techniques extend these settings to general nonlinear projection of data into low dimensionality such that the given auxiliary information is taken into account. One way to extend linear approaches to nonlinear settings is offered by kernelization. This incorporates an implicit nonlinear mapping to a high dimensional feature space together with the linear low dimensional mapping. It can be used for every linear approach which relies on dot products in the feature space only such that an efficient computation is possible, such as several variants of kernel LDA [6, 106]. However, it is not clear how to choose the kernel since its form severely influences the final shape of the visualization. In addition, the method has quadratic complexity with respect to the number of data due to its dependency on the full Gram matrix.

Another principled way to extend DR to auxiliary information is offered by an adaptation of the underlying metric which measures similarity in the original data space. The principle of learning metrics has been introduced in [123, 125]: the standard Rie-

manian metric of the given data manifold 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 self-organizing maps (SOM), multidimensional scaling (MDS), and a recent information theoretic model for data visualization which directly relies on the metric in the data space [123, 125, 163]. A drawback of the proposed method is its high computational complexity due to the dependency of the metric on path integrals or approximations thereof. A slightly different approach is taken in [43]: Instead of learning the metric, an ad hoc adaptation is used which also takes given class labelling into account. The corresponding metric induces a k-nearest neighbour graph which is shaped according to the given auxiliary information. This can directly be integrated into a supervised version of Isomap. The principle of discriminative visualization by means of a change of the metric is considered in more generality in the approach [21]. Here, a metric induced by prototype based matrix adaptation as introduced e.g. in [134, 135] is integrated in several popular visualization schemes including Isomap, manifold charting, locally linear embedding, etc.

Alternative approaches to incorporate auxiliary information is to modify the cost function of DR tools to include the given class information. The approaches introduced in [77, 109] can both be understood as extensions of stochastic neighbour embedding (SNE). SNE tries to minimize the deviation of the distribution of data induced by pairwise distances in the original data space and projection space, respectively. Parametric embedding (PE) substitutes these distributions by conditional probabilities of classes, given a data point, this way mapping both, data points and class centres at the same time. For this procedure, however, an assignment of data to unimodal class centres needs to be known in advance. Multiple relational embedding (MRE) incorporates several dissimilarity structures in the data space induced by labelling, for example, into one latent space representation. For this purpose, the difference of the distribution of each dissimilarity matrix and the distribution of an appropriate transform of the latent space are accumulated, whereby the transform is adapted during training according to the given task. The weighting of the single components is taken according to the task at hand, whereby the authors report an only mild influence of the weighting on the final outcome. It is not clear, however, how to pick the form of the transformation to take into account multimodal classes.

Coloured maximum variance unfolding (MVU) incorporates auxiliary information into MVU by substituting the raw data which is unfolded in MVU by the combination of the data and the covariance matrix induced by the given auxiliary information. This way, differences which should be emphasized in the visualization are weighted by the differences given by the prior labelling. Like MVU, however, the method depends on the full Gram matrix and is computationally demanding, such that approximations have to be used.

These approaches constitute promising candidates which emphasize the relevance of discriminative nonlinear dimensionality reduction. Only few of these methods allow an easy extension to new data points or approximate inverse mappings. Further, most methods suffer from high computational costs which make them infeasible for large data sets. In this context, GTM poses a promising approach to be extended for discriminative

DR. It defines an explicit mapping from low-dimensional to high-dimensional space with the possibility to compute the inverse projection and is quite fast with only linear time complexity.

3.2 The generative topographic mapping

The GTM [12], see also [146] for a detailed derivation, provides a generative stochastic model of data $\mathbf{x} \in \mathbb{R}^D$ which is induced by a mixture of Gaussians with the means given by the points $\mathbf{u}$ positioned on a regular grid in the low-dimensional latent space. These are mapped to prototypical target vectors in the high-dimensional data space

$$\mathbf{u} \mapsto \mathbf{x} = y(\mathbf{u}, \mathbf{W}), \tag{3.1}$$

where the function y is parametrized by $\mathbf{W}$. This is shown schematically on the Figure 3.1. Typically a generalized linear regression model is chosen

$$y : \mathbf{u} \mapsto \Phi(\mathbf{u}) \cdot \mathbf{W} \tag{3.2}$$

induced by the base functions Φ, such as equally spaced Gaussians with variance σ^{-1}. Every latent point induces a Gaussian distribution

$$p(\mathbf{x}|\mathbf{u}, \mathbf{W}, \beta) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left(-\frac{\beta}{2}\|\mathbf{x} - y(\mathbf{u}, \mathbf{W})\|^2\right) \tag{3.3}$$

with variance β^{-1}, which generates a mixture of K modes

$$p(\mathbf{x}|\mathbf{W}, \beta) = \sum_{k=1}^{K} p(\mathbf{u}_k) p(\mathbf{x}|\mathbf{u}_k, \mathbf{W}, \beta) \tag{3.4}$$

where $p(\mathbf{u}_k)$ is often chosen according to the uniform distribution, i.e. $p(\mathbf{u}_k) = 1/K$. During the training of GTM the data log-likelihood

$$\ln\left(\prod_{n=1}^{N}\left(\sum_{k=1}^{K} p(\mathbf{u}_k) p(\mathbf{x}_n|\mathbf{u}_k, \mathbf{W}, \beta)\right)\right) \tag{3.5}$$

is optimised with respect to $\mathbf{W}$ and β, where independence of the data points $\mathbf{x}_n$ is assumed. This can be done by means of an EM approach which treats the generative mixture component $\mathbf{u}_k$ for the data point $\mathbf{x}_n$ as a latent variable. Choosing a generalized linear regression model and a distribution of the latent points which is uniformly peaked at the lattice positions, EM training can be computed explicitly. It computes alternately the responsibilities

$$R_{kn}(\mathbf{W}, \beta) = p(\mathbf{u}_k|\mathbf{x}_n, \mathbf{W}, \beta) = \frac{p(\mathbf{x}_n|\mathbf{u}_k, \mathbf{W}, \beta) p(\mathbf{u}_k)}{\sum_{k'} p(\mathbf{x}_n|\mathbf{u}_{k'}, \mathbf{W}, \beta) p(\mathbf{u}_{k'})} \tag{3.6}$$

of component k for point number n, and the model parameters by means of the formulas

$$\mathbf{\Phi}^T \mathbf{G}_{\text{old}} \mathbf{\Phi} \mathbf{W}_{\text{new}} = \mathbf{\Phi}^T \mathbf{R}_{\text{old}} \mathbf{X} \tag{3.7}$$

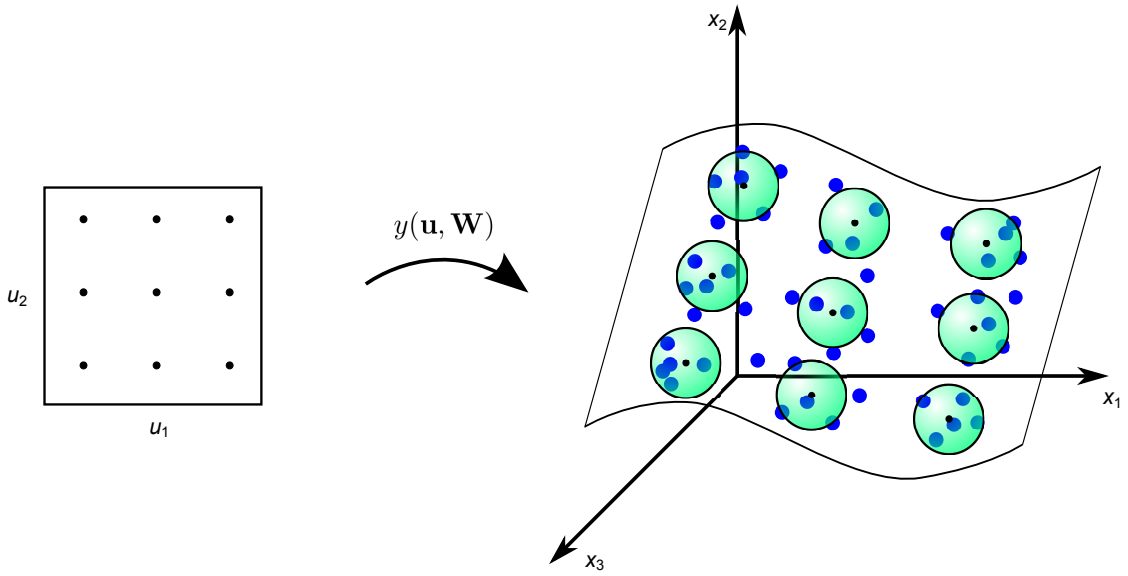


Figure 3.1: Schematic representation of the GTM. Low-dim. latent variables on a regular grid induce a Gaussian mixture model in the high-dim. data space.

for $\mathbf{W}$, where $\mathbf{\Phi}$ refers to the matrix of base functions Φ evaluated at points $\mathbf{u}_k$, $\mathbf{X}$ to the data points, $\mathbf{R}$ to the responsibilities, and $\mathbf{G}$ is a diagonal matrix with accumulated responsibilities $G_{nn} = \sum_n R_{kn}(\mathbf{W}, \beta)$. The variance can be computed by

$$\frac{1}{\beta_{\text{new}}} = \frac{1}{ND} \sum_{k,n} R_{kn}(\mathbf{W}_{\text{old}}, \beta_{\text{old}}) \|\Phi(\mathbf{u}_k)\mathbf{W}_{\text{new}} - \mathbf{x}_n\|^2 \tag{3.8}$$

where D is the data dimensionality and N the number of data points.

3.3 Relevance learning

The principle of relevance learning has been introduced in [65] as a particularly simple and efficient method to adapt the metric of prototype based classifiers according to the given situation at hand. It takes into account a relevance scheme of the data dimensionalities by substituting the squared Euclidean metric by the weighted form

$$d_{\boldsymbol{\lambda}}(\mathbf{x}, \mathbf{t}) = \sum_{d=1}^{D} \lambda_d^2 (x_d - t_d)^2 \,. \tag{3.9}$$

In [65], the Euclidean metric is substituted by the more general form (3.9) and, parallel to prototype updates, the metric parameters $\boldsymbol{\lambda}$ are adapted according to the given classification task. The principle is extended in [134, 135] to the more general metric form

$$d_{\mathbf{\Omega}}(\mathbf{x}, \mathbf{t}) = (\mathbf{x} - \mathbf{t})^T \mathbf{\Omega}^T \mathbf{\Omega} (\mathbf{x} - \mathbf{t}) \tag{3.10}$$

Using a square matrix $\mathbf{\Omega}$, a positive semi-definite matrix which gives rise to a valid pseudo-metric is achieved this way. In [134, 135], these metrics are considered in local and global form, i.e. the adaptive metric parameters can be identical for the full model, or they can be attached to every prototype present in the model. Here we introduce the same principle into GTM.

Labeling of GTM

Assume that data point $\mathbf{x}_n$ is equipped with label information l_n which is element of a finite set of different labels. By posterior labelling, GTM gives rise to a probabilistic classification of data points, assigning the label of prototype $\mathbf{t}_k$ to data point $\mathbf{x}_n$ with probability R_{kn}. Thereby, posterior labelling of GTM can be done in such a way that the classification error $\sum_{n=1}^{N}\sum_{k=1}^{K} R_{kn}(1-\delta_k(\mathbf{x}_n))$ is minimized with $\delta_k(\mathbf{x}_n)$ equal to one, if the prototype $\mathbf{t}_k$ has the same label as $\mathbf{x}_n$ and equal to zero otherwise. Thus the prototype $\mathbf{t}_k = y(\mathbf{u}_k, \mathbf{W})$ is labelled

$$c(\mathbf{t}_k) = \arg\max_c \left(\sum_{n|l_n=c} R_{kn} \right). \tag{3.11}$$

Metric Adaptation in GTM

We can introduce relevance learning into GTM by substituting the Euclidean metric in the Gaussian functions (3.3) by the more general diagonal metric (3.9) which includes relevance terms or the metric induced by a full matrix (3.10) which can also take correlations of the dimensions into account. Thereby, we can introduce one global metric for the full model, or, alternatively, we can introduce local metric parameters $\boldsymbol{\lambda}_k$ or $\mathbf{\Omega}_k$, respectively, for every prototype $\mathbf{t}_k$. We refer to the latter version as local method.

Using this posterior labelling of the prototypes, the parameters of the GTM model should be adapted such that the data log-likelihood is optimum. Analogous to [12], it can be seen that optimization of the parameters $\mathbf{W}$ and β of GTM can be done the same way as beforehand, whereby the new metric structure (3.9,3.10) has to be used when computing the responsibilities (3.6).

We assume that the metric is changed during this optimization process on a slower time scale such that the auxiliary information is mirrored in the metric parameters. Thereby, we assume quasi-stationarity of metric parameters when performing original EM training. A similar procedure has been used in [134] for simultaneous metric and prototype learning, and [142] provides an explanation in how far this procedure is reasonable in the context of self-organizing maps. Essentially, the adaptation can be understood as an adiabatic process this way [13], overlaying fast parameter adaptation by EM optimization of the log-likelihood and slow metric adaptation according to the objectives as will be detailed below.

Now the question is how to design an efficient scheme for metric learning based on the structure as provided by GTM and the given auxiliary labelling. Unlike approaches such as fuzzy-labelled SOM [133], we use a fully supervised scheme to learn metric parameters. Unlike the original framework of learning metrics [123, 125], however, we

make use of the prototype-based structure of the induced GTM classifier which allows us to efficiently update local metric parameters without the necessity of a computationally costly approximation of the Riemannian metric induced by the general Fisher information. For this purpose, we introduce two different cost functions E (see below) motivated from prototype-based learning which are used to optimize the metric parameters.

For metric adaptation, we simply use a stochastic gradient descent of the cost functions. Naturally, more advanced schemes would be possible, but a simple gradient descent already leads to satisfactory results, as we will demonstrate in experiments. To avoid convergence to trivial optima such as zero we pose constraints on the metric parameters of the form $\|\boldsymbol{\lambda}\| = 1$ or $\mathrm{trace}(\boldsymbol{\Omega}^T\boldsymbol{\Omega})^2 = 1$, respectively. This is achieved by normalization of the values, i.e. after every gradient step, $\boldsymbol{\lambda}$ is divided by its length, and $\boldsymbol{\Omega}$ is divided by the square root of $\mathrm{trace}(\boldsymbol{\Omega}^T\boldsymbol{\Omega})$. Thus, a high-level description of the algorithm is possible as depicted in Tab. 3.1. Usually, we alternate between one EM step, one epoch of gradient descent, and normalization in our experiments. Since EM optimization is much faster than gradient descent, this way, we can enforce that the metric parameters are adapted on a slower time scale. Hence we can assume an approximately constant metric for the EM optimization, i.e. the EM scheme optimizes the likelihood as before. Metric adaptation takes place considering quasi stationary states of the GTM solution due to the slower time scale.

Note that metric adaptation introduces a large number of additional parameters into the model depending on the input dimensionality. One can raise the question whether this leads to strong overfitting of the model. We will see in experiments that this is not the case: when evaluating the clustering performance of the resulting GTM, the training error is representative for the generalization error. One can substantiate this experimental finding with a theoretical counterpart: using posterior labelling, GTM offers a prototype based classification scheme with local adaptive metrics. This function class has a supervised pendant: generalized matrix learning vector quantization as introduced in [134]. The worst case generalization ability of the latter class can be investigated based on classical computational learning theory. It turns out that its generalization ability does not depend on the number of parameters adapted during training, rather, large margin generalization bounds can be derived. In consequence, very good generalization ability can be proved (and experimentally observed) as detailed in [134]. Since the formal argumentation in [134] depends on the considered function class only and not the way in which training takes place, the same generalization bounds apply to GTM with adaptive metrics as introduced here.

Now, we discuss concrete cost functions E for the metric adaptation.

Generalized Relevance GTM (GRGTM)

Metric parameters have the form $\boldsymbol{\lambda}$ or $\boldsymbol{\lambda}_k$ for a diagonal metric (3.9) and $\boldsymbol{\Omega}$ or $\boldsymbol{\Omega}_k$ for a full matrix (3.10), depending on whether a local or global scheme is considered. In the following, we define the general parameter Θ_k which can be chosen as one of these four possibilities depending on the given setting. Thereby, we can assume that Θ_k can be realized by a matrix which has diagonal form (for relevance learning) or full matrix form (for matrix updates).

Table 3.1: Integration of relevance learning into GTM

INIT
REPEAT
  E-STEP: DETERMINE R_{kn} BASED ON THE GENERAL METRIC
  M-STEP: DETERMINE W AND β AS IN GTM
  LABEL PROTOTYPES
  ADAPT METRIC PARAMETERS BY STOCHASTIC GRADIENT DESCENT OF E
  NORMALIZE THE METRIC PARAMETERS

The cost function of generalized relevance GTM is taken from generalized relevance learning vector quantization (GRLVQ), which can be interpreted as maximizing the hypothesis margin of a prototype based classification scheme such as LVQ [65, 134]. The cost function has the form

$$E(\Theta) = \sum_n E_n(\Theta) = \sum_n \mathrm{sgd}\left(\frac{d_{\Theta^+}(\mathbf{x}_n, \mathbf{t}^+) - d_{\Theta^-}(\mathbf{x}_n, \mathbf{t}^-)}{d_{\Theta^+}(\mathbf{x}_n, \mathbf{t}^+) + d_{\Theta^-}(\mathbf{x}_n, \mathbf{t}^-)}\right) \tag{3.12}$$

where $\mathrm{sgd}(x) = (1 + \exp(-x))^{-1}$, $\mathbf{t}^+$ is the closest prototype in the data space with the same label as $\mathbf{x}_n$ and $\mathbf{t}^-$ is the closest prototype with a different label.

The adaptation formulas can be derived thereof by taking the derivatives. Depending on the form of the metric, the derivative of the metric is

$$\frac{\partial d_{\boldsymbol{\lambda}}(\mathbf{x}, \mathbf{t})}{\partial \lambda_i} = 2\lambda_i (x_i - t_i)^2 \tag{3.13}$$

for a diagonal metric and

$$\frac{\partial d_{\boldsymbol{\Omega}}(\mathbf{x}, \mathbf{t})}{\partial \Omega_{ij}} = 2(x_j - t_j) \sum_d \Omega_{id}(x_d - t_d) \tag{3.14}$$

for a full matrix.

For simplicity, we denote the respective squared distances to the closest correct and wrong prototype, respectively, by $d^+ = d_{\Theta^+}(\mathbf{x}_n, \mathbf{t}^+)$ and $d^- = d_{\Theta^-}(\mathbf{x}_n, \mathbf{t}^-)$. The term sgd' is a shorthand notation for $\mathrm{sgd}'((d^+ - d^-)/(d^+ + d^-))$. Given a data point $\mathbf{x}_n$ the derivative of the corresponding summand of cost function E with respect to metric parameters yields

$$\frac{\partial E_n}{\partial \Theta^+} = 2\,\mathrm{sgd}' \cdot \frac{d^-}{(d^+ + d^-)^2} \cdot \frac{\partial d^+}{\partial \Theta^+} \tag{3.15}$$

for the parameters of the closest correct prototype and

$$\frac{\partial E_n}{\partial \Theta^-} = -2\,\mathrm{sgd}' \cdot \frac{d^+}{(d^+ + d^-)^2} \cdot \frac{\partial d^-}{\partial \Theta^-} \tag{3.16}$$

for the parameters attached to the closest wrong prototype. All other parameters are not affected. These updates take place for the local modelling of parameters, which we refer

to by local generalized relevance GTM (LGRGTM) or local generalized matrix GTM (LGMGTM), respectively. If metric parameters are global, the update corresponds to the sum of these two derivatives, referred to by generalized relevance GTM (GRGTM) or generalized matrix GTM (GMGTM), respectively.

Robust Soft GTM (RSGTM)

Unlike GRLVQ, robust soft LVQ (RSLVQ) [138] has the goal to optimize a statistical model which defines the data distribution. It is assumed that data are given by a Gaussian mixture of prototypes which are labelled. The objective is to maximize the logarithm of the probability of a data point being generated by a prototype of the correct class versus the overall probability. In the limit of small variance of the Gaussians, a learning rule which is similar to the standard LVQ rule results. The objective for a general variance β^{-1} of the Gaussian modes corresponds to the following cost function:

$$E(\Theta) = \sum_n E_n(\Theta) = \sum_n \log \left(\frac{\sum_{k|c(\mathbf{t}_k)=l_n} p(\mathbf{u}_k) p(\mathbf{x}_n|\mathbf{u}_k, \mathbf{W}, \beta)}{p(\mathbf{x}_n|\mathbf{W}, \beta)} \right) \tag{3.17}$$

Here, we can choose Gaussian modes as provided by GTM, i.e. the modes and corresponding mixture are given in analogy to formulas (3.3,3.4) where the new parametrized metric (3.9,3.10) as well as the labelling (3.11) of GTM are used.

We obtain the update rules by taking the derivatives, as beforehand:

$$\frac{\partial E_n}{\partial \Theta_k} = ((1 - \delta_k(\mathbf{x}_n))(Q_{kn} - R_{kn}) - \delta_k(\mathbf{x}_n) R_{kn}) \left(\frac{1}{S_k} \cdot \frac{\partial S_k}{\partial \Theta_k} - \frac{\beta}{2} \cdot \frac{\partial d_{\Theta_k}(\mathbf{x}_n, \mathbf{t}_k)}{\partial \Theta_k} \right) \tag{3.18}$$

where $\delta_k(\mathbf{x}_n)$ indicates whether prototype and data label coincide,

$$Q_{kn} = \frac{p(\mathbf{x}_n|\mathbf{u}_k, \mathbf{W}, \beta) p(\mathbf{u}_k)}{\sum_{k'|c(\mathbf{t}_{k'})=l_n} p(\mathbf{x}_n|\mathbf{u}_{k'}, \mathbf{W}, \beta) p(\mathbf{u}_{k'})} \tag{3.19}$$

refers to the probability of mode k among the correct modes, and

$$S_k = \left(\frac{\beta}{2\pi} \right)^{D/2} \cdot \det(\Theta_k) \tag{3.20}$$

normalizes the Gaussian modes to arrive at valid probabilities. The derivative is

$$\frac{1}{S} \cdot \frac{\partial S}{\partial \lambda_i} = \frac{1}{\lambda_i} \tag{3.21}$$

for a relevance vector and

$$\frac{1}{S} \cdot \frac{\partial S}{\partial \Omega_{ij}} = \Omega_{ji}^{-1} \tag{3.22}$$

for full matrices.

We refer to this version as local relevance robust soft GTM (LRSGTM) and local matrix robust soft GTM (LMRSGTM), respectively. The global versions can be obtained by adding the derivatives, we refer to these algorithms as relevance robust soft GTM (RSGTM) and matrix robust soft GTM (MRSGTM), respectively.

Table 3.2: Parameters used for training

data	number of prototypes	number of base functions
Landsat	10×10	4×4
Phoneme	10×10	4×4
Letter	30×30	30×30

3.4 Experiments

Classification

We test the efficiency of relevance learning in GTM on three benchmark data sets as described in [125, 163]: *Landsat Satellite data* with 36 dimensions, 6 classes, and 6,435 samples, *Letter Recognition data* with 16 dimensions, 26 classes, and 20,000 samples, and *Phoneme data* with 20 dimensions, 13 classes, and 3,656 samples. Prior to training all data sets were normalized by subtracting the mean and dividing by standard deviation. GTM is initialized using the first two principal components. The mapping $y(\mathbf{u}, \mathbf{W})$ is induced by generalized linear regression based on Gaussian base functions. The learning rate of the gradient descent for the metric parameters has been optimized for the data and is chosen in the range of 10^{-6} to 10^{-2}. More precisely, an exhaustive search of the parameter range is done and the value is picked for the learning rate which leads to the best convergence of the relevance profile. Thereby, the number of epochs is chosen as 100, which is sufficient to allow convergence of matrix parameters; typically, convergence of the EM scheme can be observed at a faster scale. The number of prototypes and base functions has been taken to suite the size of the data, it is shown in Tab. 3.2. The parameters are the same for Landsat and Phoneme. For Letter the number of prototypes is larger because there are more classes and data points. This data set is also more complex structurally, so that more base functions had to be chosen. Due to the complexity of the training, an exhaustive search of these parameters has been avoided, but reasonable numbers have been chosen. Typically, the results are only mildly influenced by small changes of these numbers. The variance of the Gaussian base functions has been chosen such that it coincides with the distance between neighboured base functions.

We report the results of a repeated stratified ten-fold cross-validation with one repeat (letter) and ten repeats (phoneme, landsat), respectively, reporting also the variance over the repeats. We evaluate the models in comparison to several recent alternative supervised visualization tools by means of the test error obtained in the cross-validation. These alternatives are taken from [163].[1] The alternative methods include parametric embedding (PE), supervised Isomap (S-Isomap), coloured maximum variance unfolding (MUHSIC), multiple relational embedding (MRE), neighbourhood component analysis (NCA), and supervised neighbourhood retrieval visualizer (SNeRV) based on different weighting of retrieval objectives in the cost function of the model (l) and a Riemannian metric based on the Fisher information matrix

The results are shown in Fig. 3.2. In two of the three cases, metric adaptation improves

[1] We would like to thank the authors of [163] for providing the results.

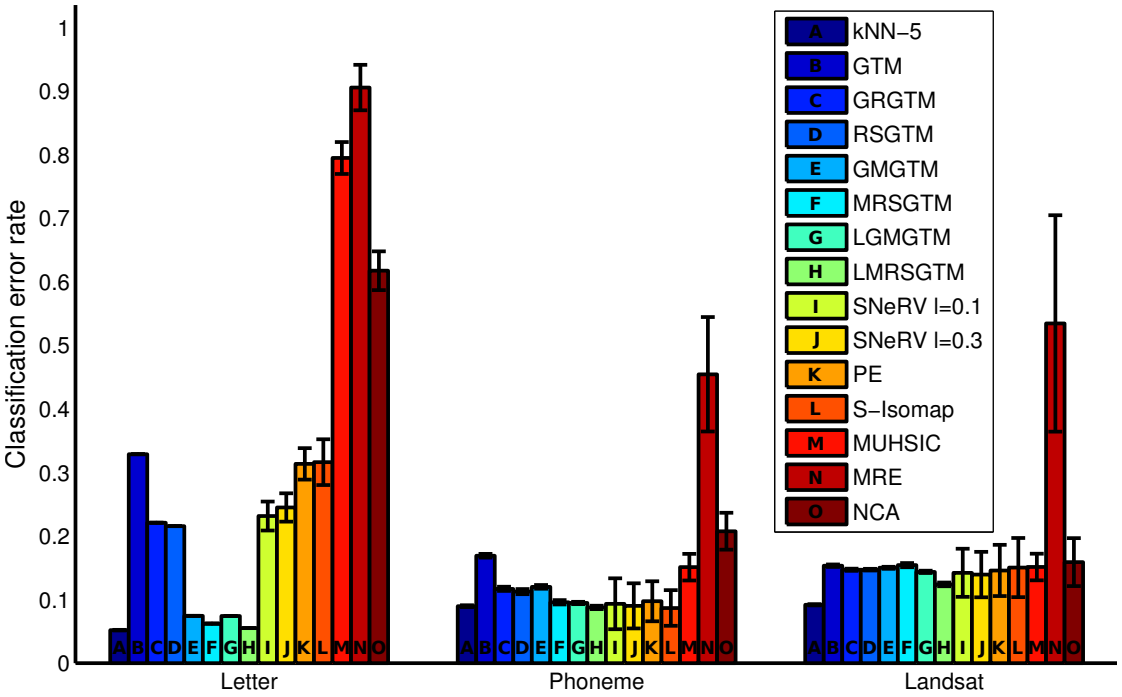


Figure 3.2: Mean accuracy of the classification obtained by diverse supervised GTM schemes as introduced in this article and alternative state-of-the-art approaches.

the classification accuracy compared to simple GTM. Thereby, matrix adaptation yields to superior results compared to the adaptation of a simple relevance vector. Further, results based on the robust soft learning vector quantization seem slightly better for all data sets. For all three data sets, we obtain state-of-the-art results which are comparable to the best alternative supervised visualization tools which are currently available in the literature. We also report the results obtained by a 5-nearest neighbour classifier for the original data in the original (high dimensional) data space. Interestingly, for all cases, the supervised results using the full information are only slightly better than the results obtained by GTM with local matrix adaptation in two dimensions. This demonstrates the high quality of the supervised visualization. Note that, unlike the experiments from [163] which restrict to a subset of 1,500 samples in all cases due to complexity issues, we can train on the full dataset due to the efficiency of relevance GTM, and, in two of the three cases, we can even perform a ten-fold repeat of the experiments in reasonable time.

Visualization

The result of a visualization of the Phoneme data set and the MNIST data set (this data set consists of 60,000 points with 768 dimensions representing the ten digits, a subsample of 6,000 images was used in this case) using robust soft GTM with local matrix adaptation are shown in Figs. 3.3 and 3.5, whereby the full data set is used for

training. A comparison to simple GTM shows the ability of matrix learning to arrive at a topographic mapping which better mirrors the underlying class structures: the pie charts display the percentage of points of the different classes assigned to the respective prototype based on the receptive fields. Interestingly, in both cases, the pie charts obtained with metric learning display less classes for the single prototypes corresponding to better separated receptive fields, whereas the classes are spread among the prototypes if metric adaptation does not take place. This is also mirrored in the better classification accuracy of GTM with matrix learning. The arrangement of the classes on the map differs for the different visualizations. For metric learning, multiple modes of the classes can be observed. For standard GTM, the distribution is less clear since the single prototypes combine different classes in their receptive fields.

One important property of GTM is the topology preservation, which can be measured with the U-matrix approach [155], computed here using the SOM toolbox [160], and is shown in Figures 3.4 and 3.6. The U-matrix allows to get insight into the shape of the map via colouring the area between the prototypes according to the distance between them. This way it is possible to see whether the prototypes lie on a smooth manifold or are heavily perturbed. For the MNIST data set, see Fig. 3.4, robust soft GTM is able learn a smoother manifold than GTM, which is indicated by brighter colors in the middle of the map, while having a few prototypes on the edges positioned away from its neighbours. In the case of Phoneme data set, see Fig. 3.6, the difference is less pronounced, which might indicate, that already GTM is able to detect the true topology of the data.

During the training, robust soft GTM learned the local metric at each prototypes position. This metric can be represented by the relevance profiles assigned to the prototypes, as shown in Figure 3.7. For the MNIST data set the profiles have a clear interpretation: since each feature represents a pixel in an image, the relevances of all features can be depicted as a grayscale picture. The dark areas correspond then to low and bright areas to high relevance. On Figure 3.7 (top) the shapes resembling digits can be recognised. They indicate which form the digits in the corresponding receptive field have, or instead show features which are important for discrimination from a neighbouring cluster. For Phoneme, Fig. 3.7 (bottom), the relevance profiles are shown as bar plots, each vertical line indicating the weighting of each feature. Interestingly, only a small deviation from the Euclidean metric can be observed, apparently it is already enough to result in a clear discrimination of the classes.

3.5 Summary

In this chapter, as noted in the Table 3.3, we proposed to integrate auxiliary information in terms of relevance updates into GTM; the benefit of this approach has been demonstrated on several benchmarks. Unlike approaches such as fuzzy-labelled SOM [133], metric parameters are adapted in a supervised fashion based on the classification ability of the model. As [125], the work is based on adaptive metrics to incorporate auxiliary information into the model. However, as already hinted in 2.4.2 the work presented in [125] is too costly to be applicable in practice. The proposed method on the other side

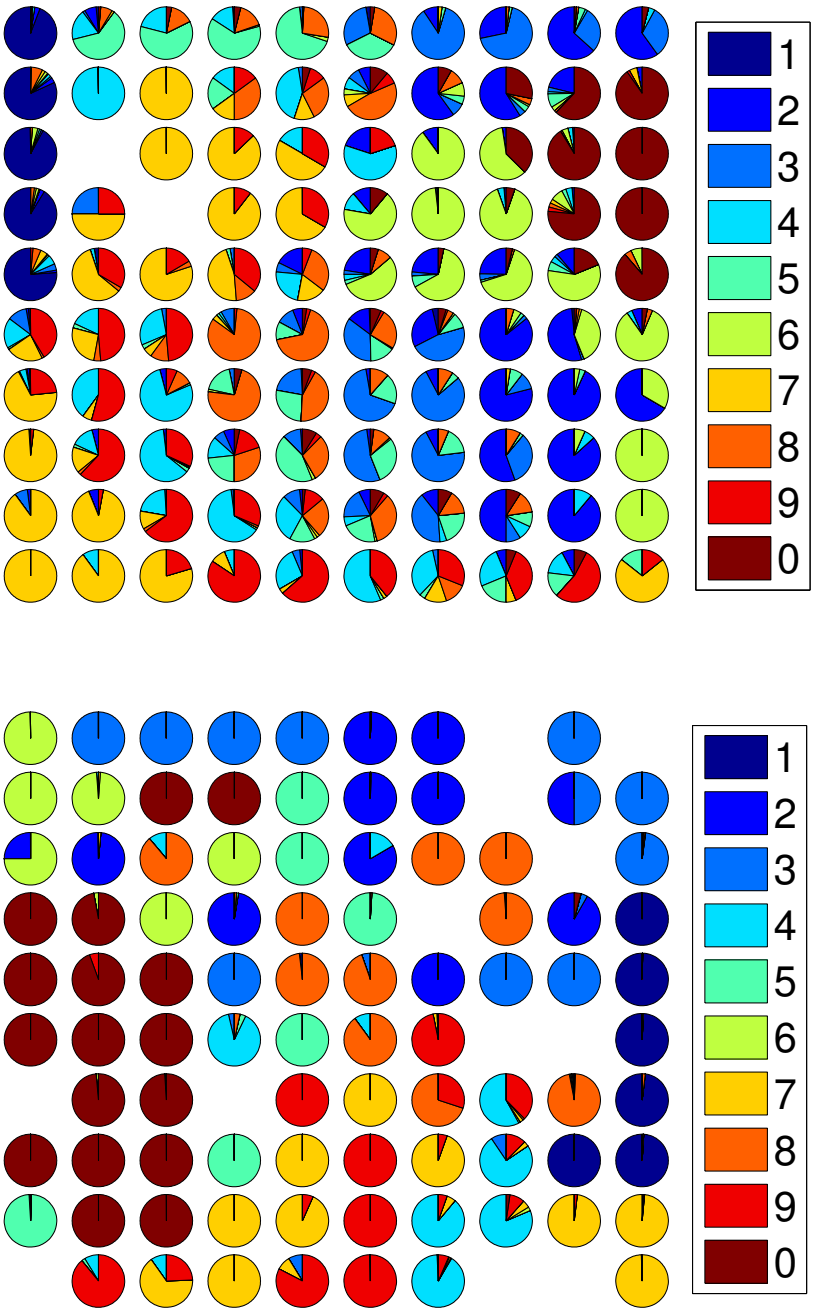


Figure 3.3: Visualization of the result of GTM (top) and robust soft GTM with local matrix learning (bottom) on the MNIST data set. Pie charts give the responsibility of the prototypes for the given classes. Supervision achieves a better separation of the classes within receptive fields of prototypes, introducing dead units if necessary.

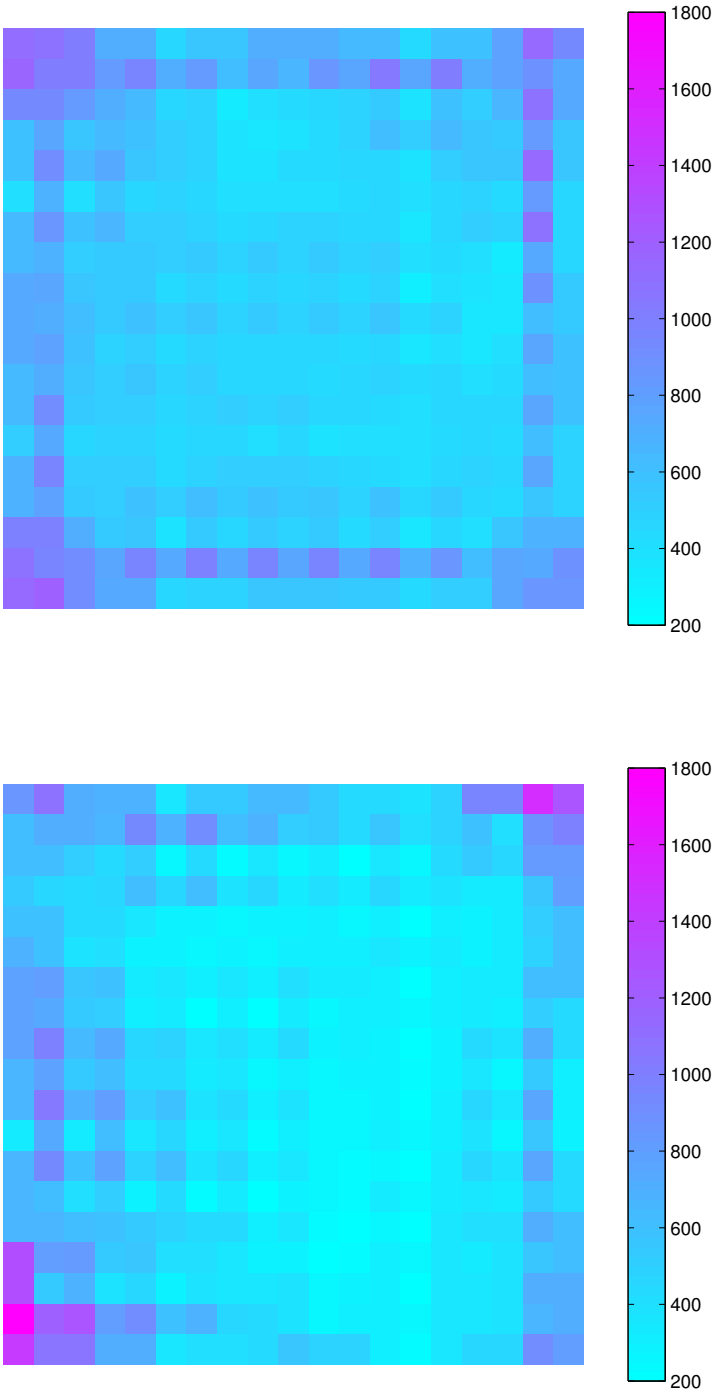


Figure 3.4: Visualization of the U-matrix for GTM (top) and robust soft GTM with local matrix learning (bottom) trained on the MNIST data set.

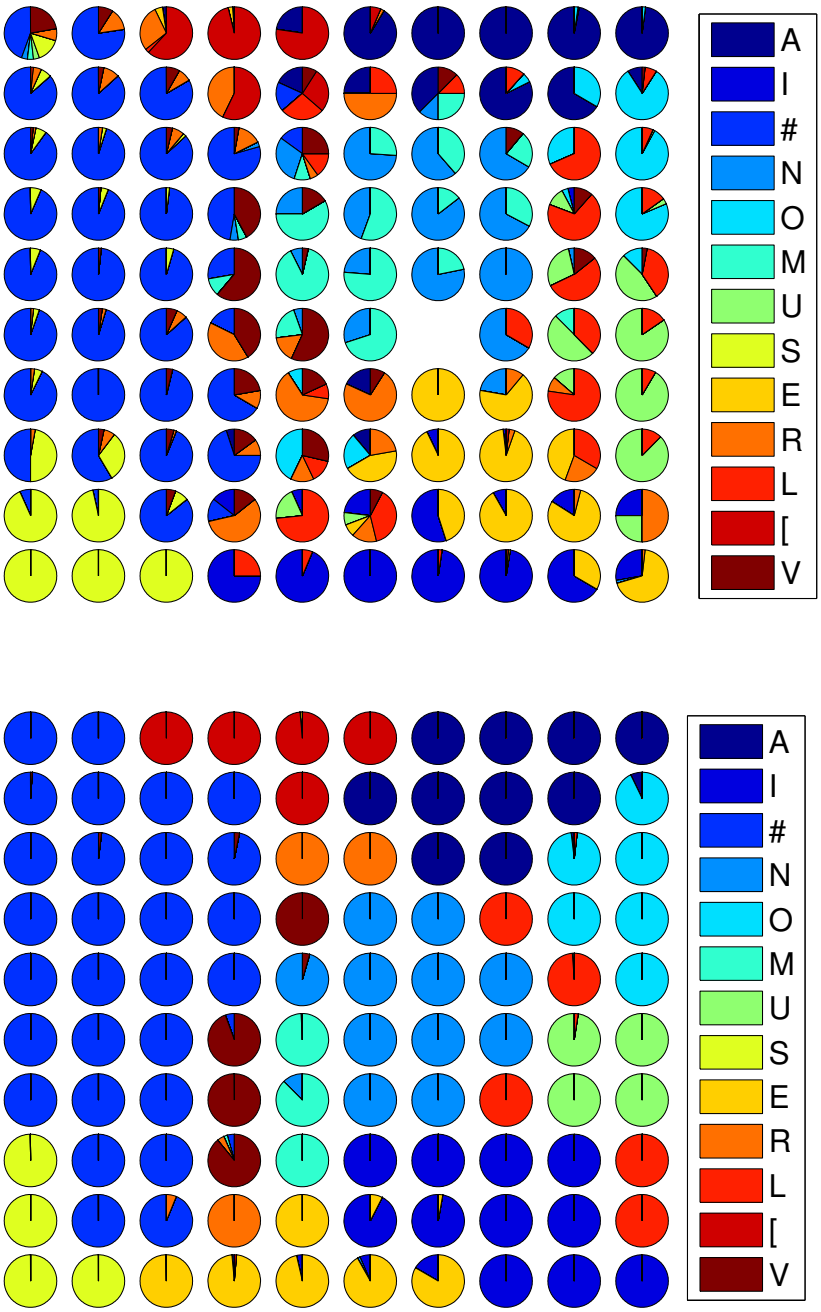


Figure 3.5: Visualization of the result of GTM (top) and robust soft GTM with local matrix learning (bottom) on the Phoneme data set. Pie charts give the responsibility of the prototypes for the given classes. Supervision achieves a better separation of the classes within receptive fields as can be seen by the pie charts.

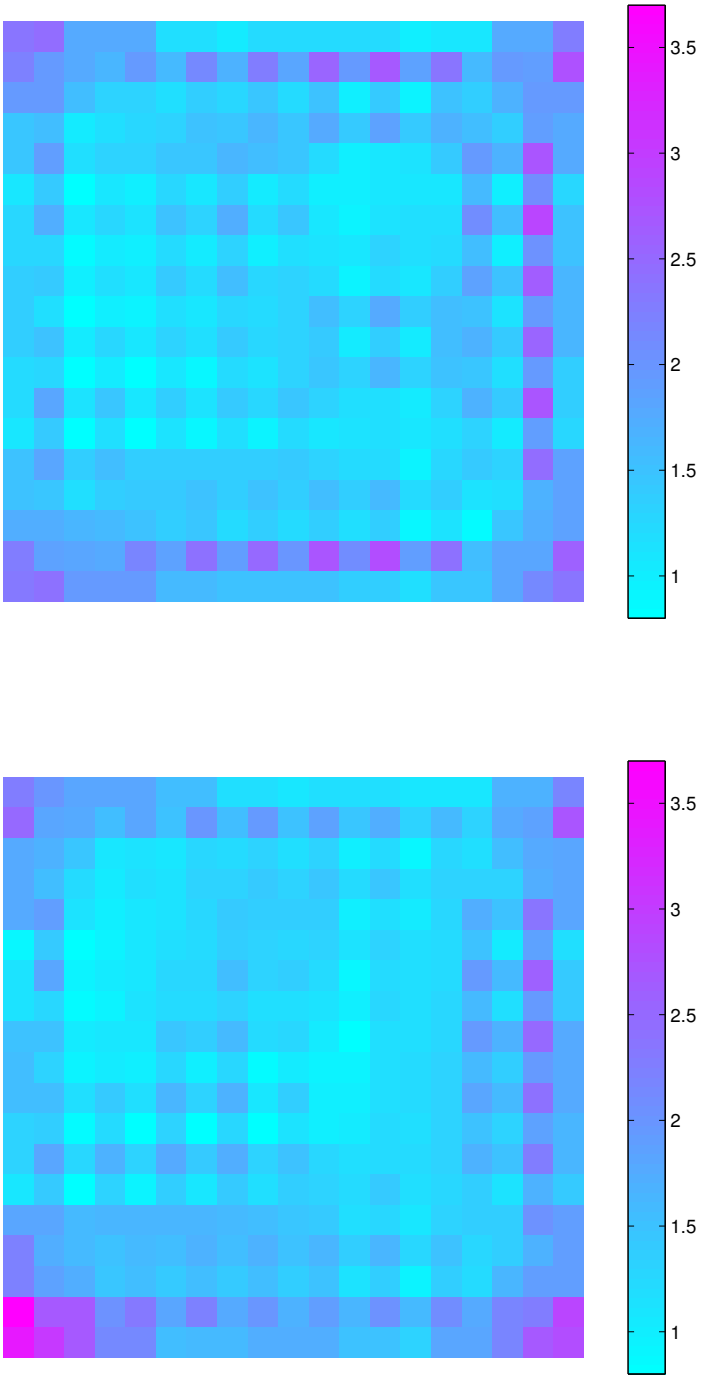


Figure 3.6: Visualization of the U-matrix for GTM (top) and robust soft GTM with local matrix learning (bottom) trained on the Phoneme data set.

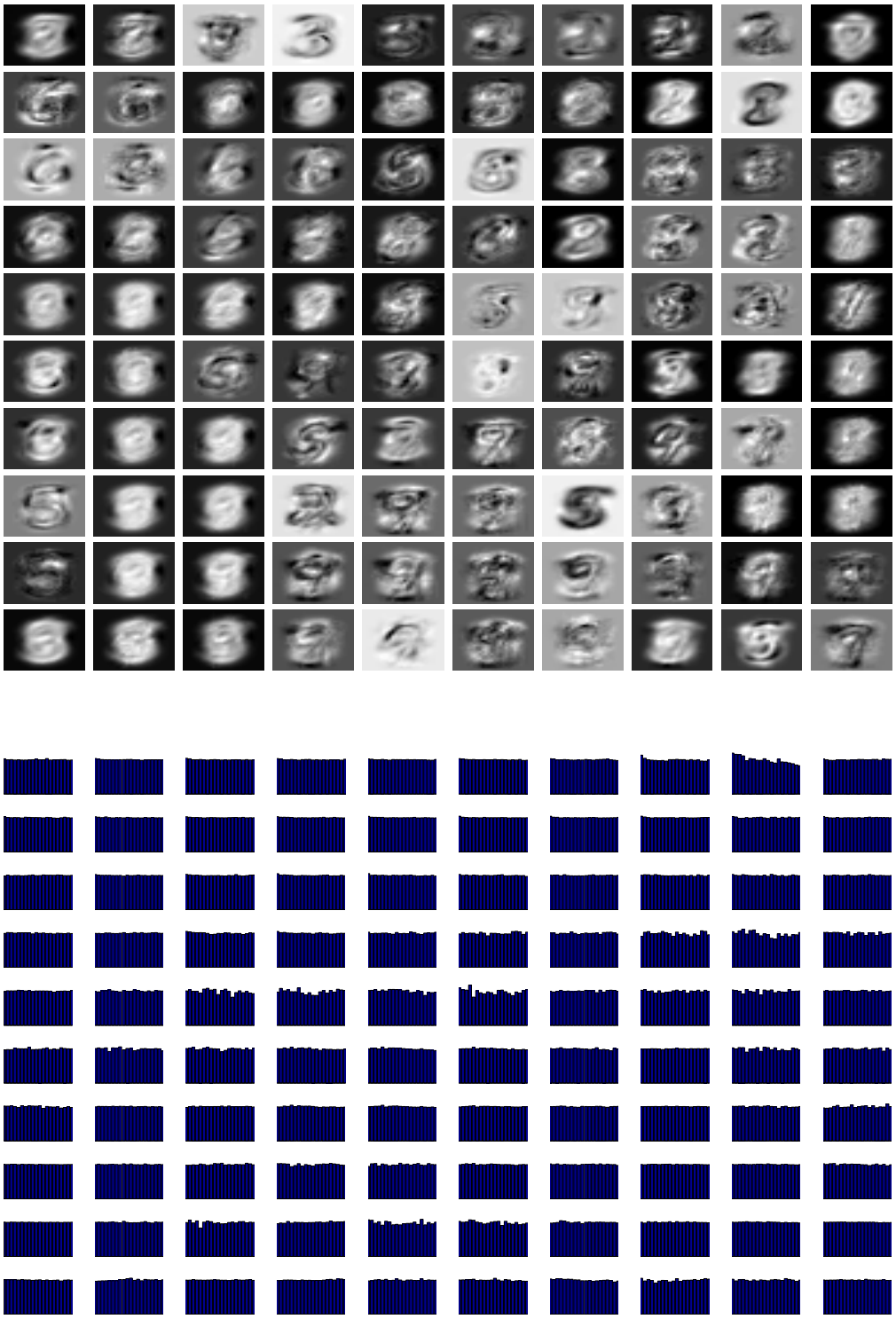


Figure 3.7: Visualization of the relevance profiles of robust soft GTM for MNIST (top) and Phoneme (bottom) data sets.

Table 3.3: Introducing relevance learning for GTM.

Topic	Relational	Out-of-Sample	Efficiency		Relevance
Technique	Data	Extension	vectorial	relational	Learning
GTM	?	√	√	?	√
t-SNE	√	?	?		?

relies on the prototype-based nature of GTM and transfers the relevance update scheme of supervised learning schemes such as [65, 134] to this setting, resulting in an efficient topological mapping. Still, the work presented in [125] is a powerful concept and we will return to it in the chapter 6.

As demonstrated on several benchmarks, the classification accuracy is competitive to state-of-the-art methods for supervised visualization, whereby GTM provides additional functionality due to the explicit topographic mapping of the latent space into the observation space accompanied by an explicit generative statistical model. As demonstrated by means of visualization, the class separation is much more accurate for supervised GTM compared to the original method, thus clearly focussing on the relevant aspects for the given classification.

This chapter is based on: Andrej Gisbrecht, Bassam Mokbel, and Barbara Hammer. Relational generative topographic mapping. *Neurocomputing*, 74(9):1359–1371, 2011.

Chapter 4

Relational generative topographic mapping

Rapidly increasing technology such as improved sensor technology and advanced methods of data preprocessing and data storage make the data more and more complex, concerning data dimensionality and information content contained in the representation. Therefore, often, a simple comparison of data in terms of the Euclidean norm and a standard representation by means of Euclidean vectors is no longer appropriate to capture the relevant aspects of the data. Rather, dissimilarity measures which are adjusted to the data type and application area at hand should be used, including, for example, alignment distances for genomic sequence analysis in bioinformatics, the compression distance to compare texts, or structure kernels to compare complex graphs and tree structures. For this reason, data mining tools which rely solely on a dissimilarity representation of data offer powerful methods for problem adapted data modelling via the canonical interface offered by the dissimilarity matrix.

In this chapter, we extend the principle of relational data processing by means of an implicit representation of prototypes to GTM. For this purpose, we use the trick of an indirect representation of prototypes in the image space in terms of linear combinations of data points and the associated possibility to compute distances in the space without an explicit reference to the vector representation of points. This way, the EM scheme of GTM can be transferred to the new setting to obtain the parameters of the model by maximizing the data log-likelihood. The efficiency and feasibility of this method, relational GTM, is demonstrated on several benchmark data sets given by dissimilarity matrices.

4.1 Related work

Classical data mining tools such as the self-organizing map (SOM) or its statistical counterpart, the generative topographic mapping (GTM) provide a sparse representation of high-dimensional data by means of latent points arranged in a low-dimensional neighbourhood structure which is useful for visualization. However, they have been introduced for Euclidean vectors only [12, 87]. Several extensions of SOM to the more general setting of data characterized by pairwise relations have been proposed, including median SOM which restricts prototype locations to data points [88], online and batch SOM using a kernelization of the classical approach [14, 175], and methods which rely on deterministic

annealing techniques borrowed from statistical physics [55]. These methods have the drawback that they can deal with discrete and restricted prototypes only (median SOM), they are restricted to kernels (kernel SOM), or they require an additional inner loop due to the necessary annealing step (deterministic annealing techniques). For specific data types such as recursive structures, the dynamics of SOM can be extended to incorporate the dependencies of data constituents. See e.g. the overviews [5, 64]. For GTM, a complex noise model as proposed in [150] allows the extension of the method to discrete structures such as sequences. Further, a kernelization of the methods is possible as described in [14, 118]. These proposals, however, are applicable to specific (recursive) data structures or kernels only.

Recently, an intuitive extension of SOM to dissimilarity data has been proposed in [67] which relies on techniques as introduced in [69]: assume that only a dissimilarity matrix characterizes the data and an explicit vectorial representation is unknown. If prototypes have the special form of convex combinations of data points, classical SOM can be computed indirectly by adapting the coefficient vectors without any explicit reference to the underlying vector space or an explicit formula of the dissimilarity measure. The resulting algorithm, relational SOM, arrives at a sparse representation of dissimilarity data in terms of virtual prototypes represented by coefficient vectors. Unlike median SOM, a continuous adaptation of prototypes is possible via the implicit representation of prototypes in terms of coefficient vectors. Interestingly, the algorithm can be interpreted as an implicit application of the SOM algorithm for an unknown vector space embedding of the underlying data, as shown in [67]. Since the algorithm relies on the dissimilarities only, this shows the invariance of the method with respect to the chosen embedding. The algorithm can be extended to an approximate iterative scheme which drastically reduces the computation time and space requirement, resulting in a linear algorithm for dissimilarity data, as proposed in [67]. This way, an efficient data mining method for very large dissimilarity data results.

SOM has the drawback that it relies on a heuristic motivation, albeit a foundation of a slightly altered version in terms of a cost function is possible [72]. The generative topographic mapping offers an alternative based on a generative statistical model [12]. It models a restricted Gaussian mixture model where the Gaussian centres are induced by a mapping of prototypes from a low-dimensional latent space. This way, visualization and sparse representation of data becomes possible. Unlike SOM, GTM training can be derived as a maximization of the data log-likelihood by an expectation maximization scheme. Further, an explicit mapping of the latent space to the data space is learned such that data can be visualized at any desired degree of granularity by choosing appropriate lattice points in the latent space.

4.2 Relational GTM

We assume that data $\mathbf{x}$ are given only indirectly in terms of pairwise dissimilarities $d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|^2$, but the vector representation $\mathbf{x}$ of the data is unknown. We assume, however, that vectors $\mathbf{x}$ exist which yield the dissimilarity matrix, albeit the corresponding vector space is not known. Therefore, the positions of the prototypes in

the data space, induced by the latent variables, are not known explicitly and the mapping

$$\mathbf{t}_k = y(\mathbf{u}_k, \mathbf{W}), \tag{4.1}$$

cannot be defined in the same way as in the vectorial case. In [69], the following fundamental observation, which allows to circumvent this issue, is presented: assume that prototypes are restricted to linear combinations of data points of the following form:

$$\mathbf{t}_k = \sum_{n=1}^{N} \alpha_{kn}\mathbf{x}_n \text{ where } \sum_{n=1}^{N} \alpha_{kn} = 1. \tag{4.2}$$

Then, the prototypes $\mathbf{t}_k$ can be represented indirectly by the means of the coefficient vector α_k. This allows to compute the dissimilarity between a data point $\mathbf{x}_n$ and a prototype $\mathbf{t}_k$ implicitly as in [69]

$$d(\mathbf{x}_n, \mathbf{t}_k) = [\mathbf{D}\alpha_k]_n - \frac{1}{2} \cdot \alpha_k^T \mathbf{D}\alpha_k \tag{4.3}$$

where $\mathbf{D}$ refers to the matrix of pairwise dissimilarities of data points and $[\cdot]_i$ is component i of the vector. It has been shown in [67], that relation (4.3) holds even for every vector space equipped with bilinear form if the targets fulfil (4.2), whereby coefficients α_{ij} must sum to 1 but they can be negative. This observation has been used in [67] to derive a relational variant of SOM. We show, that the same principle allows us to generalize GTM to relational data described by a dissimilarity matrix $\mathbf{D}$.

Thus, we assume a dissimilarity matrix $\mathbf{D}$ is given. We restrict prototype vectors $\mathbf{t}_k$ to linear combinations of data points as in (4.2). That means, the relation

$$\mathbf{T} = \alpha \cdot \mathbf{X} \tag{4.4}$$

holds where $\mathbf{T}$ denotes the target vectors, α denotes the matrix of their implicit coefficient-based representation in terms of α_k, and $\mathbf{X}$ is the matrix of observed data vectors. Note that the coefficients are not restricted to nonnegative values, since target vectors can lie outside the convex hull of the data points, i.e. $\alpha_{kn} \in \mathbb{R}$. Depending on the smoothness of the mapping of the latent space to the data space, this fact seems reasonable to arrive at a topology representing map. We can represent these prototypes indirectly in terms of coefficients α_k without any reference to an explicit vectorial representation.

Since the embedding space of $\mathbf{t}_k$ is not known, we directly treat the mapping of latent points to prototype points as a mapping of the latent space to the coefficients which represent the targets:

$$y : \mathbf{u}_k \mapsto \alpha_k = \Phi(\mathbf{u}_k) \cdot \mathbf{W} \tag{4.5}$$

where, now, $\mathbf{W} \in \mathbb{R}^{M \times N}$. This corresponds to a generalized linear regression of the latent space into the (unknown) surrounding vector space due to the linear dependency of the targets and coefficients (4.4). As before, Φ refers to M base functions such as equally spaced Gaussians with variance σ^{-1} in the latent space. In the α-space of linear combinations of data points, data points $\mathbf{x}_i$ itself are represented by unit vectors, i.e. $(0, \ldots, 0, 1, 0, \ldots, 0)$ in consequence, the data matrix $\mathbf{X}$ is now the identity matrix $\mathbf{I}$.

As before, the targets $\mathbf{t}_k$ induce a distribution in the data space given by a mixture of Gaussians centred around these points

$$p(\mathbf{x}|\mathbf{u}, \mathbf{W}, \beta) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left(-\frac{\beta}{2}\|\mathbf{x} - y(\mathbf{u}, \mathbf{W})\|^2\right). \tag{4.6}$$

The targets $\mathbf{t}_k$ are restricted to images of data points on a regular lattice in a low dimensional latent space, i.e. they are obtained via a generalized linear regression model of points $\mathbf{u}$ in latent space.

To apply (4.3), we put the restriction

$$\sum_n [\Phi(\mathbf{u}_k) \cdot \mathbf{W}]_n = 1. \tag{4.7}$$

This way, the likelihood function

$$\ln\left(\prod_{n=1}^{N}\left(\sum_{k=1}^{K} p(\mathbf{u}_k)p(\mathbf{x}_n|\mathbf{u}_k, \mathbf{W}, \beta)\right)\right) \tag{4.8}$$

can be computed based on (4.6) where the distance computation can be performed indirectly using (4.3).

As for GTM, we can use an EM optimization scheme to arrive at solutions for the parameters β and $\mathbf{W}$, where, again, the mode $\mathbf{u}_k$ responsible for data point $\mathbf{x}_n$ serves as latent variable. As in the vectorial case, an EM algorithm in turn computes the responsibilities

$$R_{kn}(\mathbf{W}, \beta) = p(\mathbf{u}_k|\mathbf{x}_n, \mathbf{W}, \beta) = \frac{p(\mathbf{x}_n|\mathbf{u}_k, \mathbf{W}, \beta)p(\mathbf{u}_k)}{\sum_{k'} p(\mathbf{x}_n|\mathbf{u}_{k'}, \mathbf{W}, \beta)p(\mathbf{u}_{k'})} \tag{4.9}$$

using now the alternative formula for the distances (4.3), and it optimizes the expectation

$$\sum_{k,n} R_{kn}(\mathbf{W}_{\text{old}}, \beta_{\text{old}}) \ln p(\mathbf{x}_n|\mathbf{u}_k, \mathbf{W}_{\text{new}}, \beta_{\text{new}}) \tag{4.10}$$

with respect to $\mathbf{W}$ and β under the constraint (4.7). This latter problem reads as

$$\max\, l(\mathbf{W}) := \ln\left(\prod_{n=1}^{N}\left(\sum_{k=1}^{K} p(\mathbf{u}_k)p(\mathbf{x}_n|\mathbf{u}_k, \mathbf{W}, \beta)\right)\right) \tag{4.11}$$

subject to

$$g_k(\mathbf{W}) := \sum_n [\Phi(\mathbf{u}_k) \cdot \mathbf{W}]_n - 1 = 0. \tag{4.12}$$

This is a constrained optimization problem and can be solved using the method of Lagrange multipliers. The corresponding Lagrangian function is

$$\Lambda(\mathbf{W}, \mu) := l(\mathbf{W}) + \sum_k \mu_k g_k(\mathbf{W}), \tag{4.13}$$

where μ_k are the Lagrange multipliers. The optimum of this function can be derived by solving $\nabla_{\mathbf{W},\mu}\Lambda = 0$. This leads to the equations

$$\nabla_{\mathbf{W}}\Lambda = 0 \Leftrightarrow \beta\mathbf{\Phi}^T\mathbf{G}\mathbf{\Phi}\mathbf{W} = \beta\mathbf{\Phi}^T\mathbf{R}\mathbf{I} - \mathbf{\Phi}^T\mu\mathbf{1}_N^T \tag{4.14}$$

and

$$\nabla_{\mu}\Lambda = 0 \Leftrightarrow \mathbf{\Phi}\mathbf{W}\mathbf{1}_N = \mathbf{1}_K, \tag{4.15}$$

where $\mathbf{\Phi}$ refers to the matrix of the base functions Φ evaluated at points $\mathbf{u}_k$, $\mathbf{R}$ to the responsibilities, and $\mathbf{G}$ is a diagonal matrix with accumulated responsibilities $G_{nn} = \sum_n R_{kn}(\mathbf{W}, \beta)$. By left multiplying (4.15) with $\beta\mathbf{\Phi}^T\mathbf{G}$ we get

$$\beta\mathbf{\Phi}^T\mathbf{G}\mathbf{\Phi}\mathbf{W}\mathbf{1}_N = \beta\mathbf{\Phi}^T\mathbf{G}\mathbf{1}_K$$

A substitution with (4.14) leads to

$$\left(\beta\mathbf{\Phi}^T\mathbf{R}\mathbf{I} - \mathbf{\Phi}^T\mu\mathbf{1}_N^T\right)\mathbf{1}_N = \beta\mathbf{\Phi}^T\mathbf{G}\mathbf{1}_K \tag{4.16}$$

$$\beta\mathbf{\Phi}^T\left(\mathbf{R}\mathbf{I}\mathbf{1}_N - \mathbf{G}\mathbf{1}_K\right) = \mathbf{\Phi}^T\mu N. \tag{4.17}$$

In the equations (4.14) to (4.17) $\mathbf{I}$ refers to a representation of the data points in the space of linear combinations, it is the identity matrix. In consequence, the left side vanishes. Because of the linear independence of the base functions Φ, μ must be a zero vector. Thus, it follows that the standard solution of this cost function without constraints automatically fulfils the given constraints. Hence the model parameters can be determined in analogy to (3.7,3.8) where, now, functions Φ map from the latent space to the space of coefficients α, i.e. we solve

$$\mathbf{\Phi}^T\mathbf{G}_{\text{old}}\mathbf{\Phi}\mathbf{W}_{\text{new}} = \mathbf{\Phi}^T\mathbf{R}_{\text{old}}\mathbf{I} \tag{4.18}$$

for the weights and we calculate

$$\frac{1}{\beta_{\text{new}}} = \frac{1}{ND}\sum_{k,n} R_{kn}(\mathbf{W}_{\text{old}}, \beta_{\text{old}})d\left(\Phi(\mathbf{u}_k)\mathbf{W}_{\text{new}}, \mathbf{x}_n\right) \tag{4.19}$$

for the variance where we use (4.3) to compute the dissimilarity. Here, D denotes the intrinsic dimensionality of the space of coefficients. This is upper bounded by the number of data points but in general smaller. It has to be estimated based on the given data set. In practice, setting D to a larger value or even the upper bound N does hardly affect the result of the method. We refer to this iterative update scheme as *relational GTM (RGTM)*. The pseudocode of the full algorithm is shown as Algorithm 1.

Initialization

Original GTM is initialized based on a principal component analysis (PCA) to avoid convergence to local optima. The risk of convergence to local optima would be large, otherwise, due to the rapid convergence of the EM scheme. PCA is applied to the matrix of vectorial data points and yields eigenvalues $\mathbf{e}$ and eigenvectors $\mathbf{A}$ of the covariance

matrix, which correspond to the first two principal components of the data. The Euclidean GTM is initialized by solving

$$\mathbf{\Phi W} = \mathbf{U}\mathbf{A}^T, \tag{4.20}$$

where the left hand side denotes the nonlinear mapping of the latent points to the data space and the right hand side denotes the linear projection of latent points $\mathbf{U} = (\mathbf{u}_i)_i$ to the two primary components $\mathbf{A}$. To obtain an appropriate scaling of the grid, the eigenvectors $[\mathbf{A}]_i$ are multiplied with the square roots of their eigenvalues e_i and the latent points $\mathbf{U}$ are normalized to have zero mean and standard deviation one. Afterwards the mean of the data, which is removed by the PCA, is added to the linear component of $\mathbf{W}$.

A similar principle can be applied to RGTM: In the case of RGTM we can obtain the first two principal components of the data points which are given only indirectly by using multidimensional scaling (MDS). MDS is applied to the dissimilarity matrix and yields matrix $\mathbf{A}$, which N rows are two dimensional representations of N data points. The columns of $\mathbf{A}$ denote the two principle components of the data, given as the linear combination of the data points. Based on this observation, the initialization of RGTM is done via (4.20), where, now, the left hand side denotes the nonlinear mapping of the latent points to the space of the coefficients, i.e. affine combinations, and the right hand side denotes the linear projection of the latent points $\mathbf{U}$ to the two primary components of the data in the affine space. To obtain the same scaling as in the vectorial case, the latent points $\mathbf{U}$ are normalized as above and the columns of the matrix $\mathbf{A}$ are multiplied by their standard deviation and divided by the corresponding eigenvalues of $\mathbf{A}\mathbf{A}^T$.

The data points in the space of affine combinations lie on the hyperplane, which has the distance $\sqrt{1/N}$ from the origin. Since MDS removes the mean of the data, the resulting mapped manifold contains the origin. It should be shifted, to match the data. This can be achieved by adding $\mathbf{1}/N$ to the linear component of $\mathbf{W}$.

4.3 Convergence of RGTM

Euclidean case

RGTM has been derived under the assumption that a vector space exists with data points $\mathbf{x}_i$ such that the dissimilarities can be expressed as $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2$. Under this assumption, instead of performing GTM in the unknown vector space, RGTM optimizes the data log-likelihood implicitly in the space of coefficient vectors α_i which induce prototypes $\mathbf{t}$ in the vector space by a linear combination $\mathbf{T} = \alpha \cdot \mathbf{X}$, $\mathbf{T}$ being the matrix of prototype vectors in the (unknown) data space. The procedure of RGTM is equivalent to original GTM in the data space due to the following reasons:

- The constraint $\sum_n \alpha_{kn}$ is automatically fulfilled for solutions of GTM. Therefore, because of the equality of the distances (4.3) there is a one-one correspondence of target vectors found by GTM and coefficient vectors found by RGTM.

- The solution found by RGTM depends on the model for y. We can choose it as generalized linear regression model for GTM and for RGTM. Since targets and

Algorithm 1 Relational GTM

function RelGTM($\mathbf{D}$)
  generate the grid of latent points $\{\mathbf{u}_k\}, k = 1, \ldots, K$
  prepare the generalized linear regression model
  init $\mathbf{W}$, using MDS
  init β
  compute $\alpha_k = [\mathbf{\Phi}]_k \mathbf{W}$
  compute **Dist** where $d(\mathbf{x}_n, \mathbf{t}_k) = [\mathbf{D}\alpha_k]_n - \frac{1}{2} \cdot \alpha_k^T \mathbf{D}\alpha_k$
  for $i = 1$: epochs **do**
    compute $\mathbf{R}$ from (3.6) using **Dist** and β
    compute $\mathbf{G}$ where $G_{nn} = \sum_n R_{kn}$
    compute $\mathbf{W} = \left(\mathbf{\Phi}^T \mathbf{G} \mathbf{\Phi}\right)^{-1} \mathbf{\Phi}^T \mathbf{R}$
    compute $\alpha_k = [\mathbf{\Phi}]_k \mathbf{W}$
    compute **Dist** where $d(\mathbf{x}_n, \mathbf{t}_k) = [\mathbf{D}\alpha_k]_n - \frac{1}{2} \cdot \alpha_k^T \mathbf{D}\alpha_k$
    compute $\beta = ND \left(\sum_{k,n} R_{kn} Dist_{kn}\right)^{-1}$
  end for
  return $\mathbf{\Phi}$, $\mathbf{W}$ and β
end function

coefficient vectors are linearly dependent, these two choices correspond to each other.

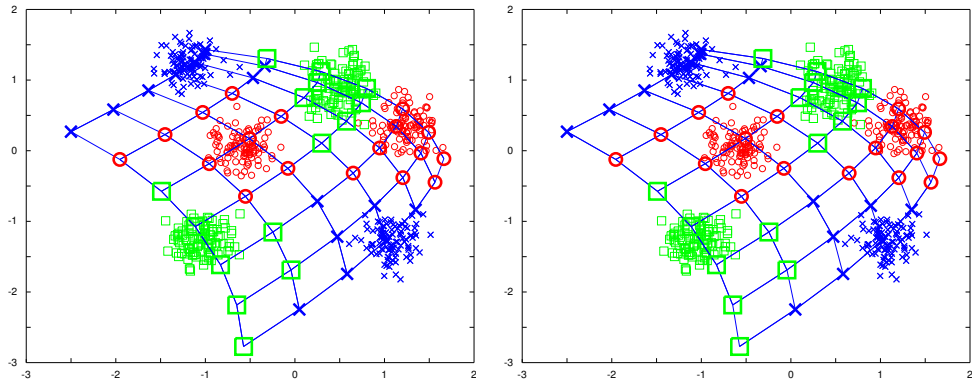
- PCA initialization of weights $\mathbf{W}$ based on the data points $\mathbf{X}$ corresponds to an MDS initialization of weights according to the dissimilarity matrix $\mathbf{D}$.
- The variance β^{-1} is adapted using the dimensionality of the data. If the intrinsic dimensionality is known, then the variance is computed in the same way for RGTM.

Therefore, if an Euclidean embedding of data exists, convergence of RGTM is guaranteed, and the procedure implicitly optimizes the data log-likelihood of the underlying (unknown) data space. GTM and RGTM yield the same results. We demonstrate this fact in a simple example involving a two-dimensional mixture of Gaussians (see Fig. gtm-rgtm). The results of GTM and RGTM are exactly identical as can be seen in Figure 4.1.

Pseudo-Euclidean case

In general, a Euclidean embedding of an arbitrary dissimilarity matrix $\mathbf{D}$ need not exist. We assume that $\mathbf{D}$ has zero diagonal $d_{ii} = 0$ and symmetric entries $d_{ij} = d_{ji}$. In this case a so-called pseudo-Euclidean embedding in a vector space can be found as explained e.g. in [119], see also section 5.2.2. That means, one can find a vector space with a bilinear form. This form need not be positive definite, but there can exist negative eigenvalues, more precisely p components are positive, q are negative. In formulas, the bilinear form is given as

$$\langle \mathbf{x}, \mathbf{y} \rangle_{p,q} = \sum_{i=1}^{p} x_i y_i - \sum_{i=p+1}^{q} x_i y_i \tag{4.21}$$



(a) GTM (b) RGTM

Figure 4.1: Comparison of GTM and RGTM on an Euclidean toy data set. The grid was plotted in the original data space. Obviously, for Euclidean data, the results are identical.

In this setting, we can find vectors $\mathbf{x}_i$ in pseudo-Euclidean space with the property $d_{ij} = \langle \mathbf{x}_i - \mathbf{x}_j, \mathbf{x}_i - \mathbf{x}_j \rangle_{p,q}$. The bilinear form needs not correspond to a positive semidefinite form, the number of negative contributions q in (4.21) referring to the necessary corrections of the data to achieve Euclideanity. Data are Euclidean iff $q = 0$.

In this general setting, RGTM can in principle be applied as beforehand since all operations of RGTM are defined. In fact, all operations of GTM being vector space operations (and thus being well defined also in pseudo-Euclidean space), RGTM corresponds to an application of the GTM algorithm in the pseudo-Euclidean vector space also for this general setting – however, without the guarantee that the data log-likelihood is optimized by this procedure. In fact, a few things can happen:

- The distance as computed by (4.3) can become negative due to the negative eigenvalues of the bilinear form. Then the corresponding probability (3.3) does not constitute a valid probability. Experimental observations indicate that this situation happens in practical experiments, but it does not seem to harm the result. The mathematical counterpart of this operation, however, is not clear, one major problem being that an appropriate mathematical definition of probability measures in pseudo-Euclidean space does not yet exist [119]. The problem of non-Euclideanity and, in consequence, no well-defined probability, could be cured by a transformation of the negative parts of pseudo-Euclidean space, operations such as flipping negative eigenvalues, clipping negative eigenvalues, or performing a spread transformation having been proposed in the literature [25, 129]. However, important information can be lost this way and results which incorporate the full information can be better, as demonstrated in [67, 91].

- β as computed in (4.19) can become negative. In this case, numerical problems occur apart from the fact that a negative variance does no longer correspond to a valid probability measure. Although this is a theoretical possibility, we never observed this behaviour in practice.

- The algorithm can diverge since it does not optimize the log-likelihood by an EM-scheme. Like its deterministic counterpart, the relational SOM, this setting can be observed in theoretical model situations [67]. It is due to the fact that the weight matrix as computed by (3.7) can correspond to a saddle point of the maximization step. However, as also reported for its deterministic counterpart [67], we never observed this behaviour for any real-life data set due to the fact that the positive parts of the pseudo-Euclidean space usually outweigh contributions due to the negative eigenvalues.

Thus, it seems possible to safely use RGTM also for general dissimilarity data sets, albeit a clear mathematical foundation in terms of a likelihood optimization is so far not available in this case.

We will discuss relational data in more detail in chapter 5. The experimental results presented there imply, that an eigenvalue correction, which would solve the problems presented here, can lead to results comparable to the state of the art. However, it is still not clear which effects occur by modifying the underlying space.

Complexity

The memory complexity of the original GTM algorithm is $\mathcal{O}(KN)$, where K is the number of latent points. This is due to the storage of the pairwise distances of prototypes and data points, as well as the corresponding responsibilities. The computational complexity of PCA, which is used for the initialization, is $\mathcal{O}(ND)$ for a D-dimensional data set and projection to two dimensions [131]. The most demanding task in training is the computation of the distances, which is $\mathcal{O}(KN)$. The matrix inversion necessary for computing $\mathbf{W}$ is cubic in the number of basis functions, which is small for a small amount of basis functions. Thus, the overall complexity of GTM is linear in the number of data points.

RGTM requires more memory than GTM. In addition to the distances and responsibilities, the larger parameter matrix $\mathbf{W}$ and the coefficients α have to be stored. These matrices are $\mathcal{O}(MN)$ and $\mathcal{O}(KN)$ respectively, where M denotes the number of basis functions. Still, the memory complexity stays linear. In RGTM, instead of PCA, MDS is used for initialization. Its computational complexity is $\mathcal{O}(N^3)$, but since we project into 2D, only the first two components are needed, which can be computed in $\mathcal{O}(N^2)$. Since the distances are calculated using the dissimilarity matrix and coefficients, the complexity becomes $\mathcal{O}(KN^2)$. Hence, the overall computational complexity of RGTM per epoch is dominated by $\mathcal{O}(N^2)$ for large data sets. That means, while RGTM extends the applicability of GTM to settings where pairwise dissimilarities rather than vectors are given, it pays the prize of an increased quadratic instead of linear complexity. This is still more efficient than an explicit embedding of the dissimilarity data into a vector space which would be $\mathcal{O}(N^3)$.

In addition, this effort is comparable to the effort of current state of the art visualization techniques for dissimilarity data. The computational complexity of t-distributed stochastic neighbour embedding (t-SNE) as one of the most popular visualization tools [158] is $\mathcal{O}(N^2)$ per epoch, being a gradient method for a cost function involving $\mathcal{O}(N^2)$

terms. The memory requirement is dominated by the embedding coefficients, i.e. $\mathcal{O}(N)$. Thus, assuming K is constant, the requirements of RGTM and t-SNE match.

In the recent time the focus of DR shifted towards big data, so that more efficient DR techniques were developed. The Barnes-Hut technique was applied to t-SNE and similar algorithms [157, 174], resulting in methods with computational complexity of $\mathcal{O}(N \log(N))$. Also, RGTM was accelerated to linear time using the Nyström approximation, which is a popular technique to speed up kernel methods. An investigation of the Nyström method for dissimilarities in the context of RGTM can be found in [47] and will be discussed more generally in chapter 5.

4.4 Experiments

Evaluation on benchmark data sets

First, we test RGTM on several benchmark dissimilarity data sets as introduced in [25, 59]:

- **Cat cortex data:** The cat cortex data originates from anatomic studies of cats' brains. The dissimilarity matrix displays the connection strength between **65** cortical areas [55]. For our purposes, a preprocessed version as presented in [57] was used. The matrix is symmetric with zero diagonal, but the triangle inequality does not hold. The data is labeled with **four classes**.

- **Protein data:** The protein data set, as described in [110], consists of **226** globin proteins which are compared based on their evolutionary distance. The samples originate from different protein families: hemoglobin-α, hemoglobin-β, myoglobin, etc. Here, we distinguish **five classes** as proposed in [57]: HA, HB, MY, GG/GP, and others. Unlike the other data sets considered here, the protein data set has a highly unbalanced class structure, with class distribution HA (31.86%), HB (31.86%), MY (17.26%), GG/GP (13.27%), and others (5.75%).

- **Aural sonar data:** The aural sonar data set, as described in [25], consists of **100** returns from a broadband active sonar system, which are labelled in **two classes**, target-of-interest versus clutter. The dissimilarity is scored by two independent human subjects each resulting in a dissimilarity score in $\{0, 0.1, \ldots, 1\}$.

- **Patrol data:** The patrol data set describes **241** members of seven patrol units and one class corresponding to people not in any unit, i.e., **eight classes**. Dissimilarities are computed based on every person in the patrol units naming five other persons in their unit, whereby the responses were partially inaccurate. Every mentioning yields an entry of the dissimilarity matrix, see [25]. Data are sparse in the sense that most entries of the matrix correspond to the maximum dissimilarity which we set to 3.

- **Voting data:** The voting data set describes a **two-class** classification problem incorporating **435** samples which are given by 16 categorical features with 3 different

possible values each. The dissimilarity is determined based on the value difference metric, see [25].

If necessary, the data sets were linearly transformed from similarities to dissimilarities prior to training. Also, in case the dissimilarities were not symmetric, we symmetrized the dissimilarity matrix $\mathbf{D}$ by setting $\tilde{\mathbf{D}} = (\mathbf{D} + \mathbf{D}^T)/2$. Diagonal values were set to zero, ignoring any self-dissimilarities: $\tilde{d_{ii}} = 0 \; \forall i \in \{1 \dots N\}$.

For one data point, we refer to the nearest prototype in data space as the *winner*. In case of RGTM, the winner for a certain point is therefore the latent point with the highest responsibility with respect to the data point. Since all benchmark data sets are labelled, it is possible to evaluate the clustering result by the classification accuracy obtained by posterior labelling. For RGTM, one can choose different labelling strategies depending on the application context: We can use standard labelling given by a majority vote as usually done for crisp approaches such as self organizing maps or neural gas. As an alternative, we can rely on the averaged responsibilities of prototypes for data R_{kn} and label prototypes according to accumulated responsibilities. Here we used majority voting to ensure comparability to neural gas and similar: a latent point is assigned the label which the majority of data points in its receptive field carries, these are all data points for which it is the winner.

We report the results of a repeated cross-validation with ten repeats, where we used 2 folds for the cat cortex data and aural sonar data and 10 folds for the other data sets to maintain comparability with the results from [59]. For the cross-validation, out-of-sample extensions of the assignments can be computed the same way as for relational neural gas, see [59]. To classify out-of-sample data, we assigned the class label of the closest prototype in data space that does carry a class label. The classification accuracy obtained on the respective test set is listed in Table 4.1. For comparison, we report the classification accuracy of deterministic annealing (DA) and relational neural gas (RNG) as presented in [59]. In the RGTM, we used 900 latent points (a 30-by-30 regular grid) and 4 Gaussian base functions (a 2-by-2 grid) for all data sets. The amount of base functions implies the degree of freedom of the manifold in data space, to which the latent points are mapped to. The number of base functions was generally chosen as small as possible to preserve the topology of the data. The target manifold therefore has a low degree of freedom, so a certain amount of unlabelled prototypes are to be expected, since they are mapped to locations with no data in their receptive fields. This fact justifies the use of more prototypes in RGTM in comparison with the experiments performed with RNG in [59], the latter not being restricted by topological constraints. The variance of the base functions, σ^{-1}, has been chosen such that it fits the distance between neighbouring base function centres. This parameter setting was chosen with regard to all data sets.

The classification accuracy on the test set and the corresponding standard deviation is reported in Tab.4.1. Obviously, the results of RGTM are comparable to these two alternatives and are even better for two of the five classification tasks. Hence, RGTM offers a feasible alternative to DA and RNG, where RGTM provides additional functionality such as topographic mapping and visualization due to an explicit modelling by means of a low-dimensional latent space.

Table 4.1: Mean classification accuracy on the data sets obtained by a repeated cross validation, the standard deviation is given in parenthesis.

	RNG	**DA**	**RGTM**
cat cortex	0.698 (0.076)	**0.803** (0.083)	0.765 (0.063)
proteins	0.919 (0.016)	0.907 (0.008)	**0.936** (0.004)
aural sonar	0.834 (0.014)	**0.856** (0.026)	0.837 (0.026)
patrol	0.665 (0.024)	0.521 (0.051)	**0.666** (0.046)
voting	0.950 (0.004)	**0.951** (0.005)	0.938 (0.006)

Visualization experiments

In Figs. 4.2 to 4.6 we show mappings of the introduced benchmark data sets obtained by RGTM in comparison with the respective visualizations by t-distributed stochastic neighbour embedding (t-SNE) as one of the currently best nonlinear data visualization techniques and the relational self-organizing map (RSOM) as an alternative model which relies on topology preservation, see [67, 87, 158]. For RGTM and RSOM we also show the corresponding U-matrices, which allow to analyse the regularity of the maps in the data space [155]. In addition to each map, we display qualitative measures for the given visualization in the graphs in Fig. 4.7.

Several quantitative evaluation measures for data visualization have recently been proposed: these include techniques which rely on neighbourhood ranking such as the trustworthiness and continuity [83, 162], which can be put into a very elegant more general framework by means of the co-ranking matrix as recently proposed in [100]. As an alternative, the contribution [163] proposes an information theoretic point of view, measuring precision and recall of local neighbourhoods induced by the low dimensional visualization as compared to the original data. In our case, the projection of data is induced by a clustering in low-dimensional space such that several data are represented by the same location in low dimensions. In this case, the evaluation measure as proposed by [100] is not applicable (see also [111]). Hence we use the information retrieval perspective on visualization, see [163].

The framework yields the mean *precision* and mean *recall* for dimensionality reduction scenarios, by evaluating errors in the proximity relationships between data points, occurring under spatial transformation. For every data point, a neighbourhood is defined in the original data space, and, correspondingly, in the visualization space. Following the information retrieval perspective, the former represents the truthful information, and the latter is viewed as the retrieval result. Their consistence can be compared in terms of true positives, false positives, and misses, which yields the basis for precision and recall. The neighbourhood of a data point is the set of all other points within a fixed radius from the respective data point, where the radius can be defined by any consistent notion of proximity. One possibility is to use a rank-based distance for the radius, i.e., define the neighbourhood set as the k nearest neighbours, breaking the ties deterministically. Therefore, the mean precision and mean recall is calculated for every possible neighbourhood radius $k \in \{1 \dots N-1\}$, where the respective mean is taken over the k-ary neighbourhoods of every data point.

Table 4.2: The topographic products for the RSOM and RGTM grids produced in the visualization experiments on the benchmark data sets, see Figs. 4.2 to 4.6.

	RSOM	**RGTM**
cat cortex	0.03285	**0.00008**
proteins	-0.03613	**0.00019**
aural sonar	-0.02585	**0.00034**
patrol	-0.14677	**0.00009**
voting	**-0.00197**	-0.00426

In our case, we obtain a visualization of data by the prescription $\mathbf{x} \mapsto \mathbf{u}_k$ such that $d(\mathbf{x}, y(\mathbf{u}_k, \mathbf{W}))$ is minimum (computed implicitly as given by (4.3)). Hence data points are displayed at the position of their winning prototype point in the grid. Note that, this way, all points in the receptive field of prototype $\mathbf{u}_k$ are displayed at the same position. As we will see later, it is possible to increase the granularity of the RGTM grid without retraining, such that a finer resolution of the grid structure would lead to a finer resolution of the representation of data. To evaluate the precision and recall, we use the definitions in [163] which are applicable to this setting as discussed in [111]. The graphs in Figure 4.7 show the measurements for every number k of nearest neighbours, ranging from 1 to $N - 1$. For both, the mean precision and mean recall, a value of 1 represents the highest, and 0 the lowest quality, while the combination of high precision and high recall is the desired situation, since it marks the highest preservation of spatial relationships.

For further evaluation, we report the topographic product, see [7], for each RGTM and RSOM visualization in Table 4.2. The topographic product constitutes an efficient measurement which approximately measures the degree of neighbourhood preservation as given by the grid structure. Here, a value of 0 refers to a perfect preservation of the map topology. To calculate the topographic product properly, only absolute values of distances between prototype positions in data space were considered, since these distances can become negative due to the non-Euclidean characteristics of the data space. Also, if the values were smaller than 10^{-7}, we reset them to this value to avoid numerical instabilities. Additionally, to investigate the topology of RGTM and RSOM in more detail we show the corresponding U-matrices in Figures 4.2 to 4.6, which were computed using the SOM toolbox [160]. The U-matrix colours the area between the prototypes according to the distance between them, whereby blue and red colors represent small and big distances respectively. This way the visualisation shows, whether the neurons in high-dimensional space are placed on a regular grid, or on a distorted one.

The parameter settings used in the experiments are listed in Table 4.3. As before, we used the majority vote for posterior labelling, and the variance of the base functions, σ^{-1}, was set such that it fits the distance between neighbouring base function centres. For the RSOM, the *initial neighbourhood range* r_0 was set to one half of the number of data points, as stated in Table 4.3. The neighbourhood range defines how much the update process of one neuron influences the neighbouring neurons in the RSOM grid, for details, see [67]. It is annealed exponentially to 0.01 during training, by calculating

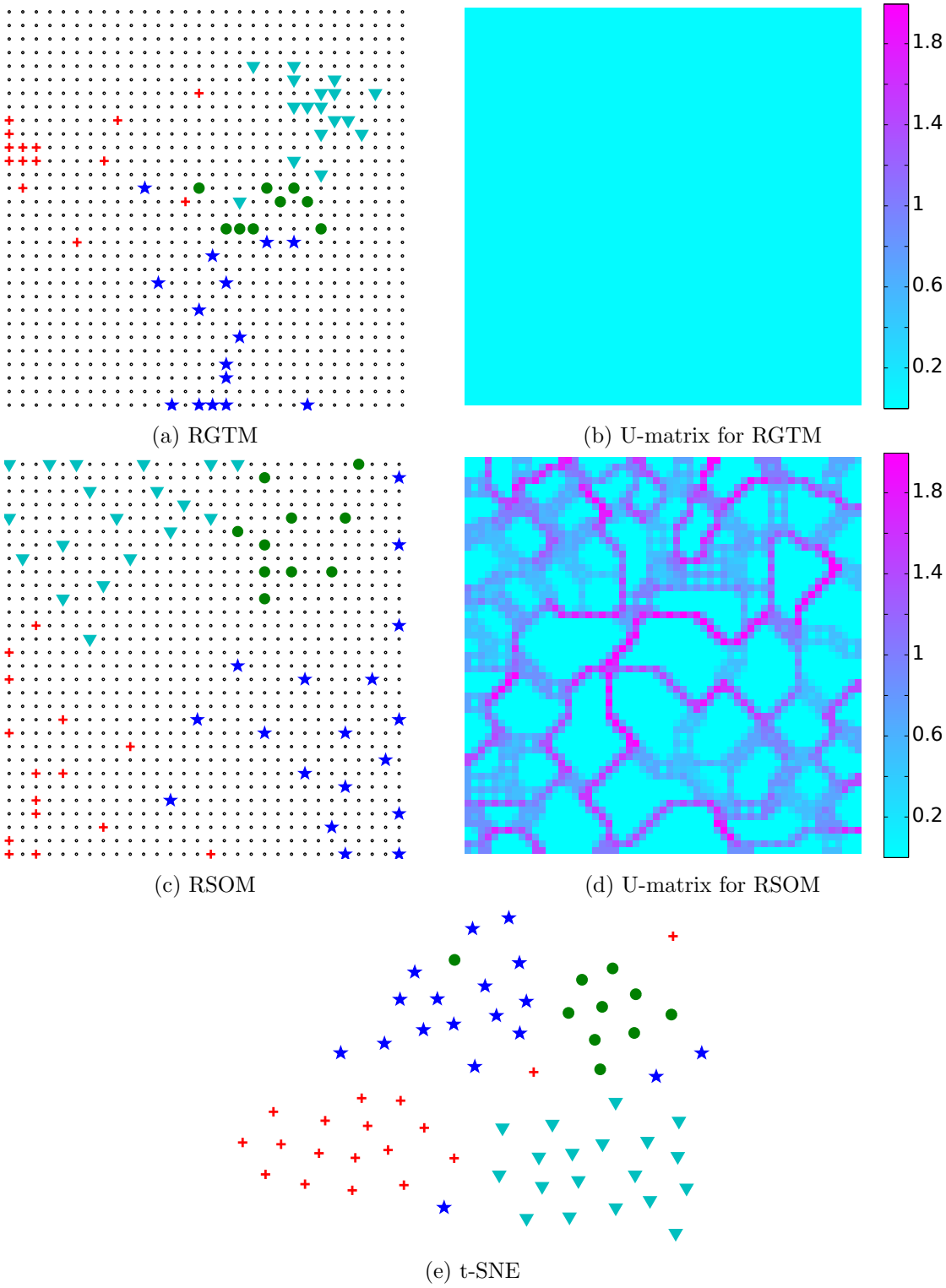
Table 4.3: Overview of the parameter settings used in the visualization experiments on the benchmark data sets.

method	**parameter**	**setting**
RGTM	number of latent points	900 (30-by-30 grid)
RGTM	number of base functions	4 (2-by-2 grid)
RGTM	number of training epochs	30
RSOM	number of neurons	900 (30-by-30 grid)
RSOM	number of training epochs	500
RSOM	initial neighbourhood range	$N/2$
t-SNE	initial dimensionality	$\lfloor N/4 \rfloor$
t-SNE	perplexity	30

the range for the current epoch e_c as $r_c = r_0 \cdot (0.01/r_0)^{e_c/e}$, where e refers to the total number of epochs. As it is common when applying t-SNE, the dimensionality of the data is first reduced by PCA, before the t-SNE mapping is calculated. This initial projection dimensionality was set to one fourth of the number of data points, as stated in Table 4.3.

As can be seen from the visualization and the evaluation in Figures 4.2 to 4.7, RGTM displays clear class structures and clear separations of the clusters in the form of unlabelled units, if appropriate. RGTM is able to preserve the topology quite well, since the flexibility of the mapping is determined by the number of base functions, which is chosen to be very small in these experiments. On the contrary, RSOM tries to optimize the quantization error in the limit, and thus spreads all prototypes among the data even at the cost of topological deformations or neighbourhood deformations, as can be seen on the visualisations of the U-matrices. They show, that RGTM has very regular grids while RSOM introduces deformations to allow clustering of the prototypes. This fact is also mirrored by the values of the topographic product as shown in Table 4.2, for which GTM yields much smaller values for all but one example. Thus, RGTM generates visualizations which are much closer to the corresponding t-SNE visualization as compared to RSOM. Unlike t-SNE, however, RGTM provides additional functionality such as grouping and an explicit lattice structure.

Interestingly, the quality as evaluated by precision and recall is much less clear. Here, RGTM leads to worse values for very small neighbourhood range k as compared to t-SNE and RSOM in almost all cases. This can be attributed to two facts: on the one hand, it clusters points such that, due to identical positions for several data points, the precision is low for small neighbourhood sizes. In addition, RGTM is quite constrained in its local projections provided a small number of base functions is chosen such that almost locally linear projections for the data manifold are observed. While this ensures an excellent global image and topology preservation as measured by the topographic product, local nonlinearities cannot be precisely captured. Due to the clustering which prevents very good values at small ranges k, the interesting region for medium sized k starting from about $k = 10$ displays a different behaviour. RGTM is at least competitive to t-SNE and RSOM, being even clearly superior to the latter in some cases, approaching the quality of t-SNE in these cases.



(a) RGTM

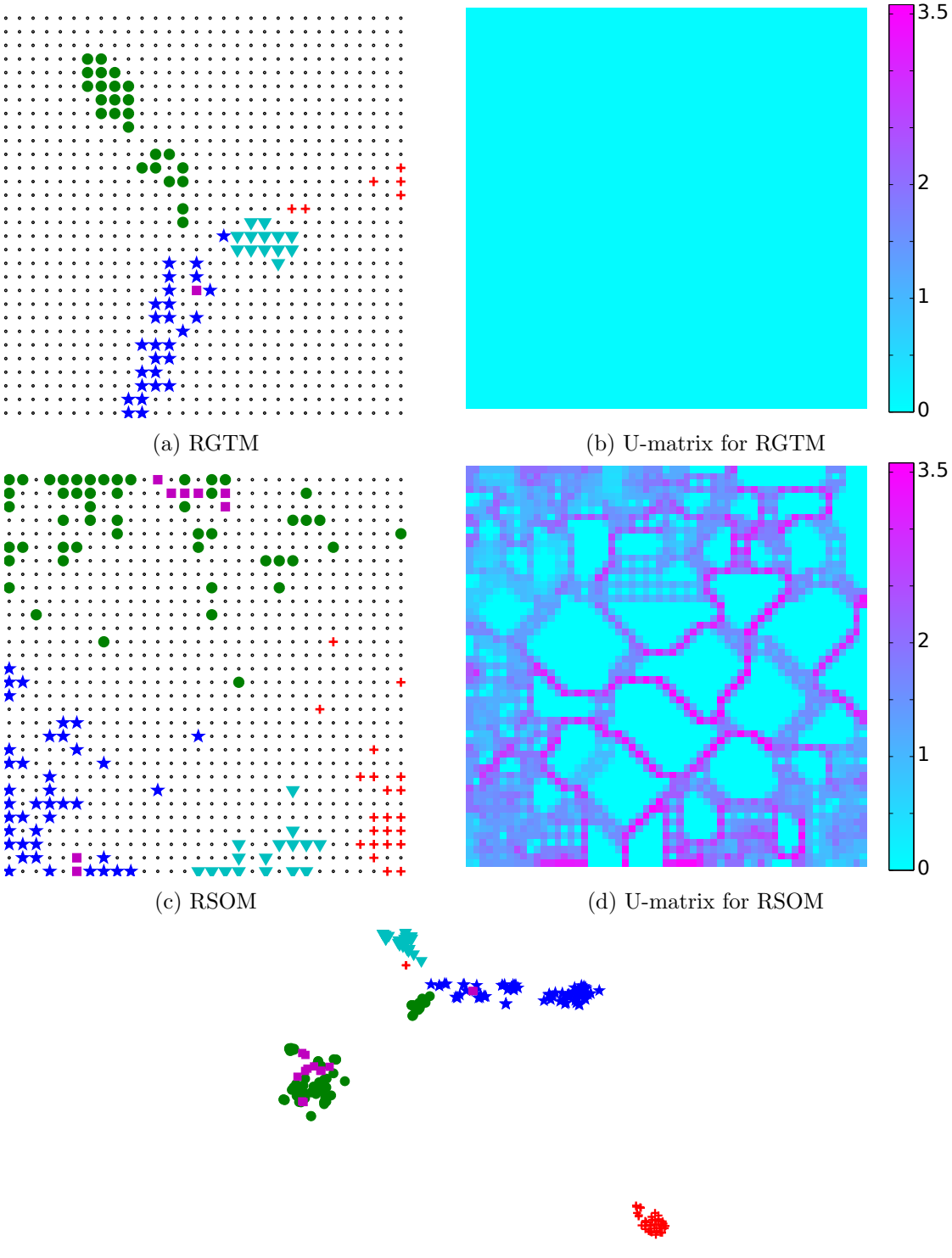
(b) U-matrix for RGTM

(c) RSOM

(d) U-matrix for RSOM

(e) t-SNE

Figure 4.2: The cat cortex benchmark data set visualized by t-SNE, as well as RGTM and RSOM with corresponding U-matrices.



(a) RGTM

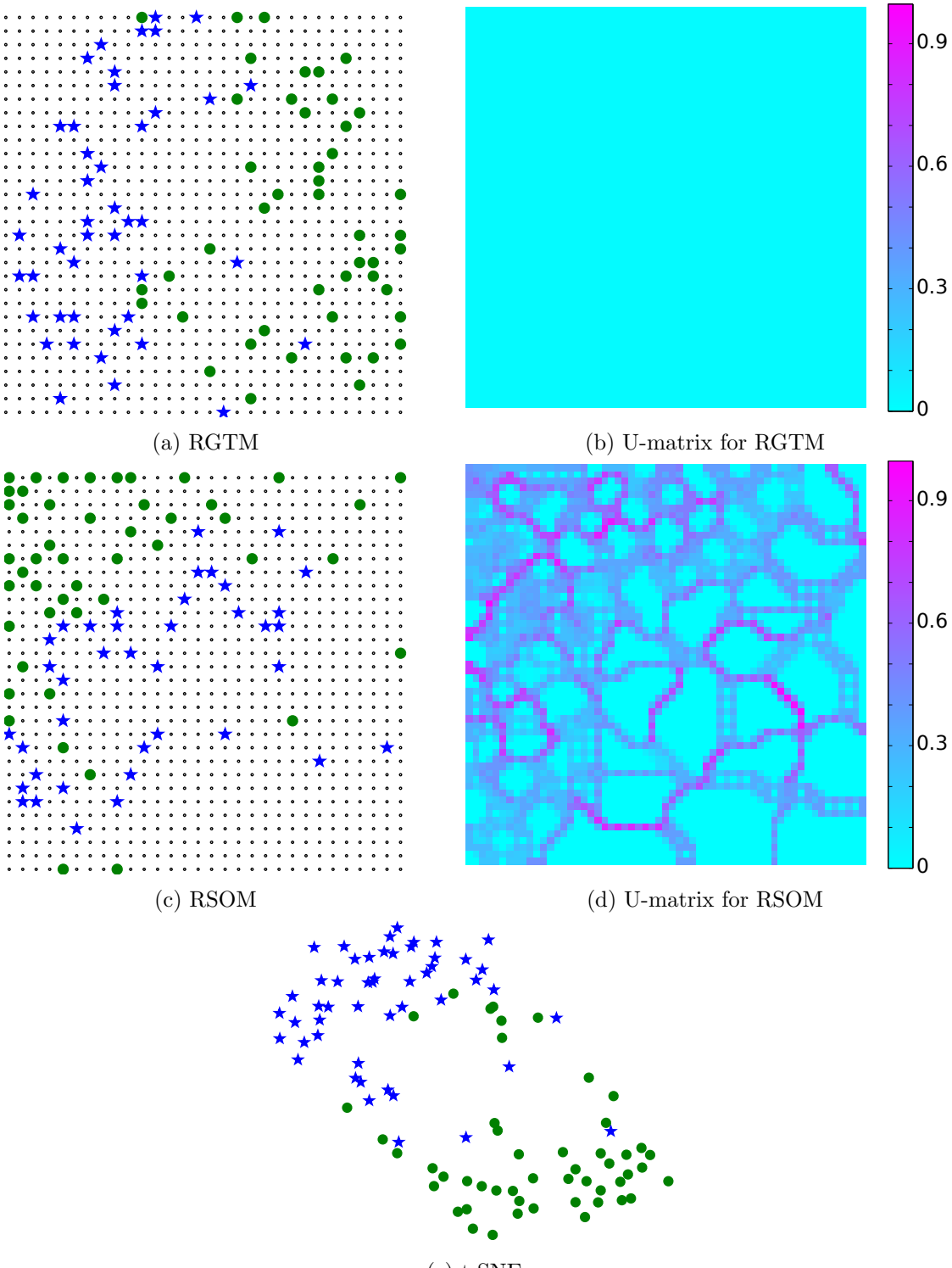
(b) U-matrix for RGTM

(c) RSOM

(d) U-matrix for RSOM

(e) t-SNE

Figure 4.3: The proteins benchmark data set visualized by t-SNE, as well as RGTM and RSOM with corresponding U-matrices.



(a) RGTM

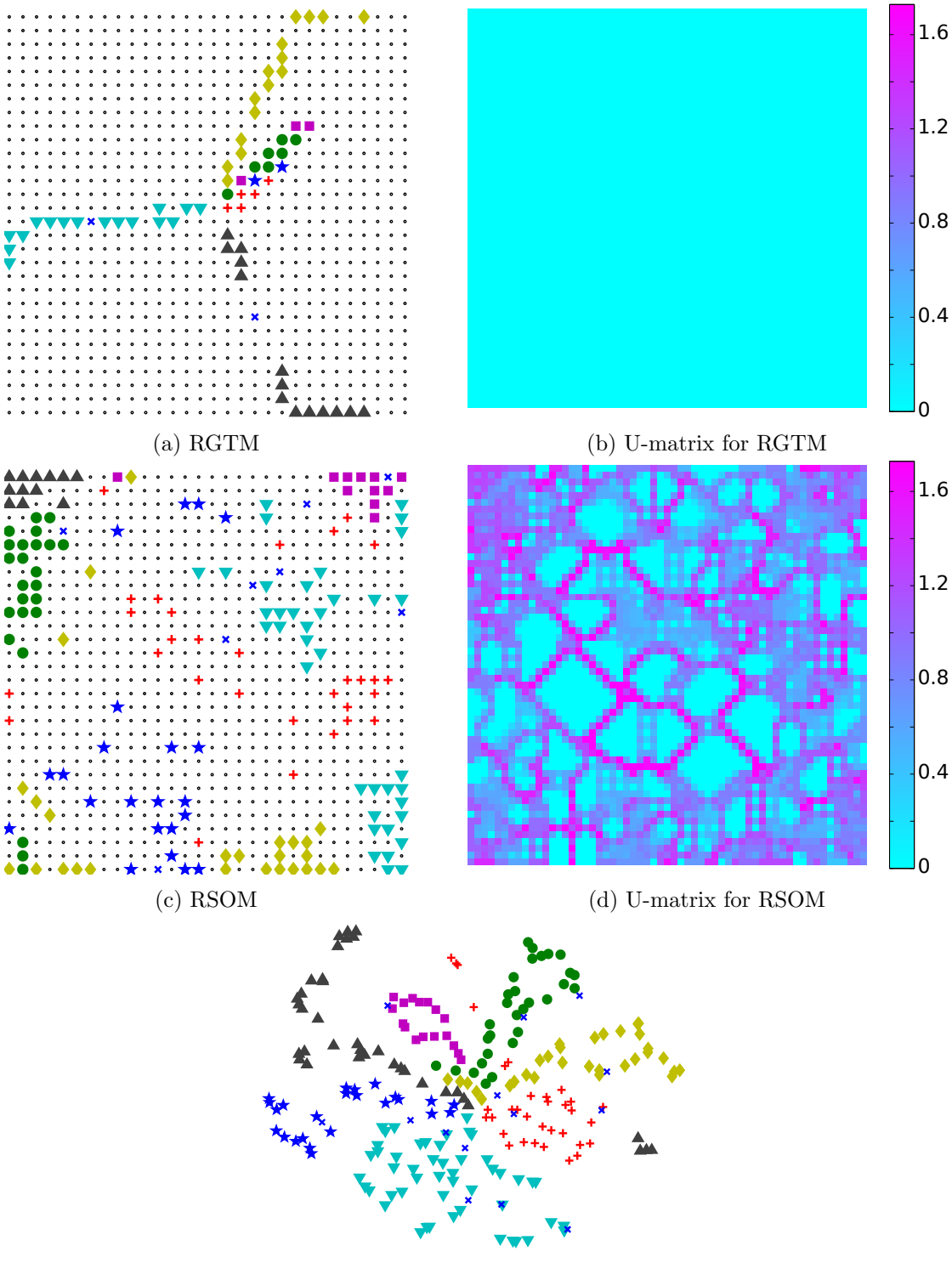
(b) U-matrix for RGTM

(c) RSOM

(d) U-matrix for RSOM

(e) t-SNE

Figure 4.4: The aural sonar benchmark data set visualized by t-SNE, as well as RGTM and RSOM with corresponding U-matrices.



(a) RGTM

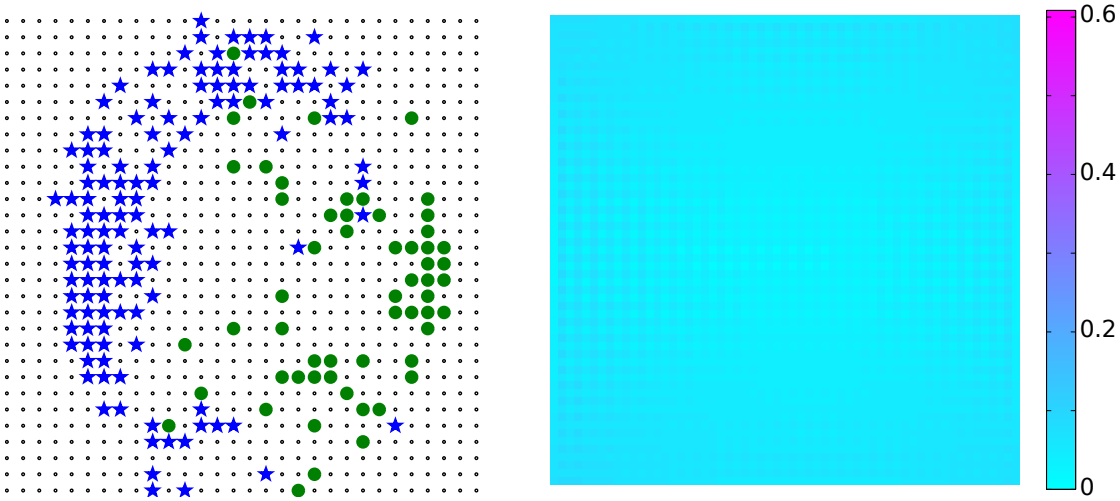
(b) U-matrix for RGTM

(c) RSOM

(d) U-matrix for RSOM

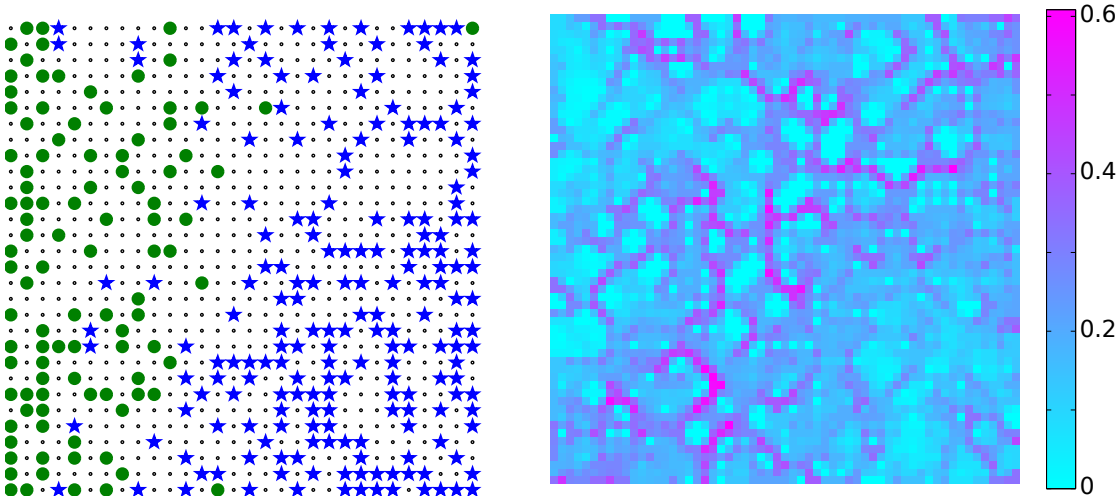
(e) t-SNE

Figure 4.5: The patrol benchmark data set visualized by t-SNE, as well as RGTM and RSOM with corresponding U-matrices.



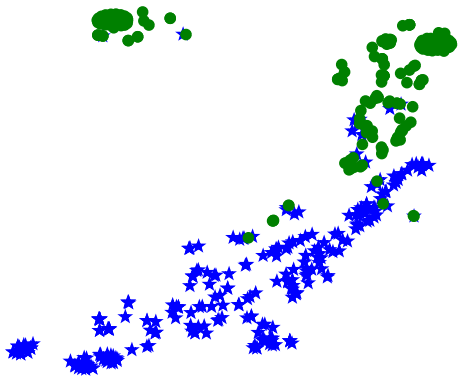
(a) RGTM

(b) U-matrix for RGTM



(c) RSOM

(d) U-matrix for RSOM



(e) t-SNE

Figure 4.6: The voting benchmark data set visualized by t-SNE, as well as RGTM and RSOM with corresponding U-matrices.

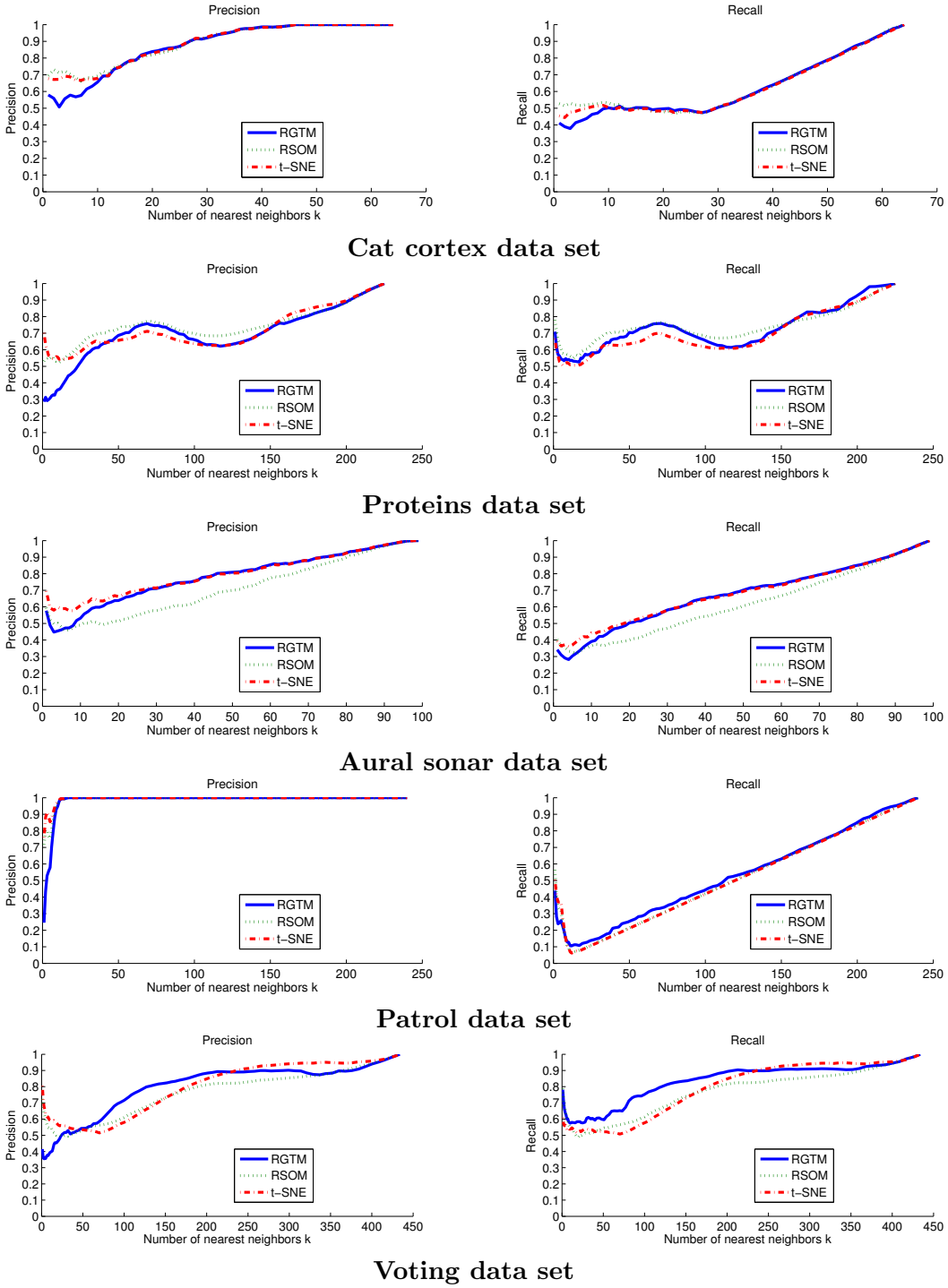


Figure 4.7: Mean precision and mean recall for the RGTM, RSOM, and t-SNE mappings of five benchmark data sets.

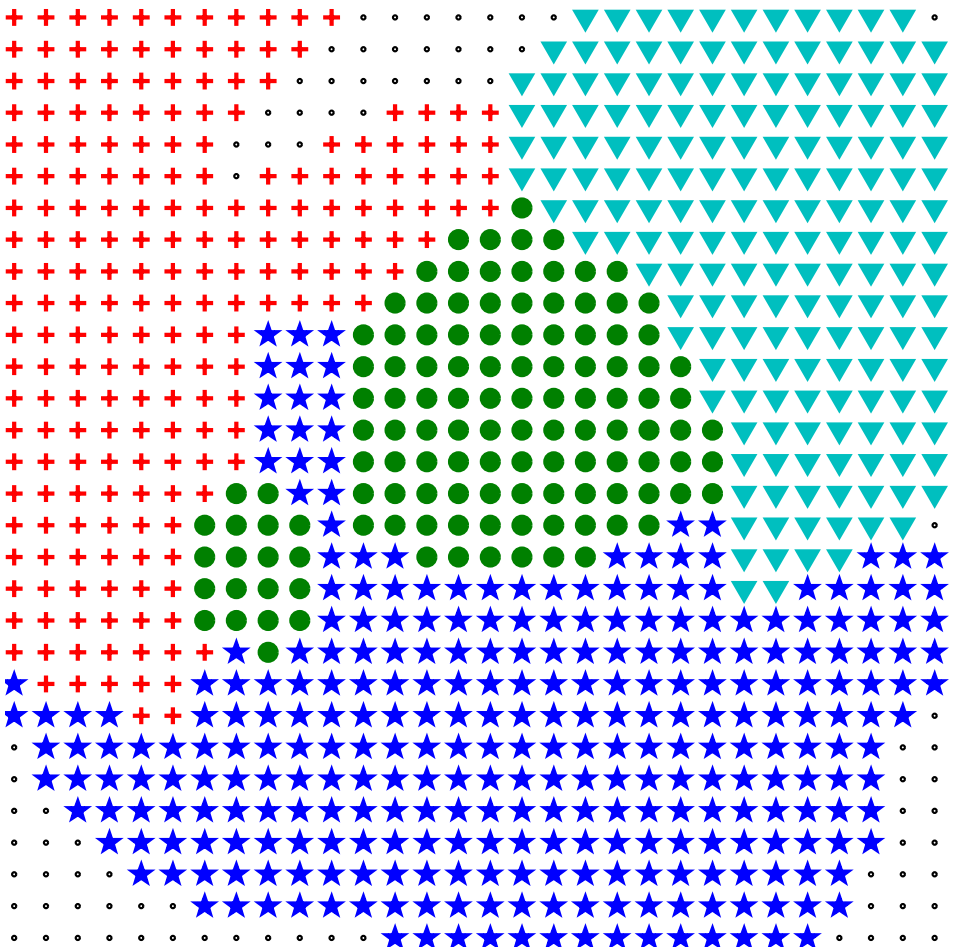
Parameters for data visualization

The parameter σ, which determines the variance of the base functions, has only a slight effect on the algorithm, if it stays in a reasonable interval. Throughout the experiments, it was set to fit the distance between neighbouring base function centres, and the number of base functions was chosen as small as possible to preserve the topology of the data. Changing the number of latent points generally changes only the sampling of the data but qualitatively the shape of the map stays the same. So with a smaller number, the algorithm is faster and sparsity of the representation is increased; with a larger number, the algorithm is slower but more details in data relations can be discovered.

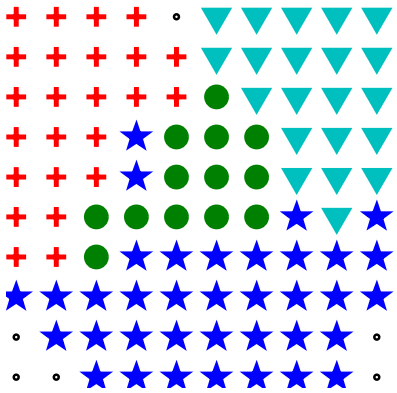
As already mentioned before, data points are projected to the grid position of its respective winning prototype in latent space. Interestingly, it is possible to use different grid resolutions for data display based on a trained map (trained using only one fixed resolution): RGTM yields an explicit mapping of latent space to the data space by means of the function (3.1). This can directly be used to create an image of any grid in latent space. This behaviour is displayed in Fig. 4.8 on the cat cortex data, trained originally with a 10-by-10 grid. Here, the grid size is changed and the trained mapping of latent points into the data space is reused to do the posterior labelling for the new grids. In this visualisation, each prototype is labelled by the class with the highest accumulated responsibility. If the highest responsibility of a prototype is below a specified threshold, then the prototype is not responsible for any class and no label is assigned to it. Obviously, the overall structure of the map remains the same, but it is possible to focus on a different level of detail on demand using an appropriate grid resolution.

By setting the grid resolution reasonably large, a very detailed resolution, in the limit a one-to-one mapping of data points to prototypes and corresponding grid positions can be achieved, i.e., all data points are individually mapped to different latent points. To favour such a mapping in an experiment, the number of latent points in the grid has to be larger than the total number of data points. Of course, depending on the topology, idle prototypes are present in the map to represent empty space. Thus, the grid resolution should be chosen as a multiple of the number of data. The standard course of action would be, to first choose a reasonable grid size, and increase the grid size, if the data are not yet represented individually. The latter is possible without retraining, relying on the explicit mapping of the latent space to the data space. Such an almost one-to-one mapping is presented for the cat cortex data in Fig. 4.9 in comparison with a t-SNE mapping. Here, a latent point is only assigned a label, if it is the winner for some data point. This way, RGTM arrives at an almost individual projection of points, rather than prototypes which deliver a compressed display of the data.

It is also possible to first train GTM with a small grid, increase the grid resolution afterwards and finally fine tune the mapping with a large grid. In this case the training with a small grid would act as an initialisation, allowing to capture the global topology of the data. The fine tuning step could then be used to construct a much more complex mapping, which would focus more on the local data structure. A similar idea is pursued by the hierarchical GTM [151]. There, a relatively small grid presents the global view on the data and each prototype can be unfolded hierarchically to a grid depicting the local data structure.



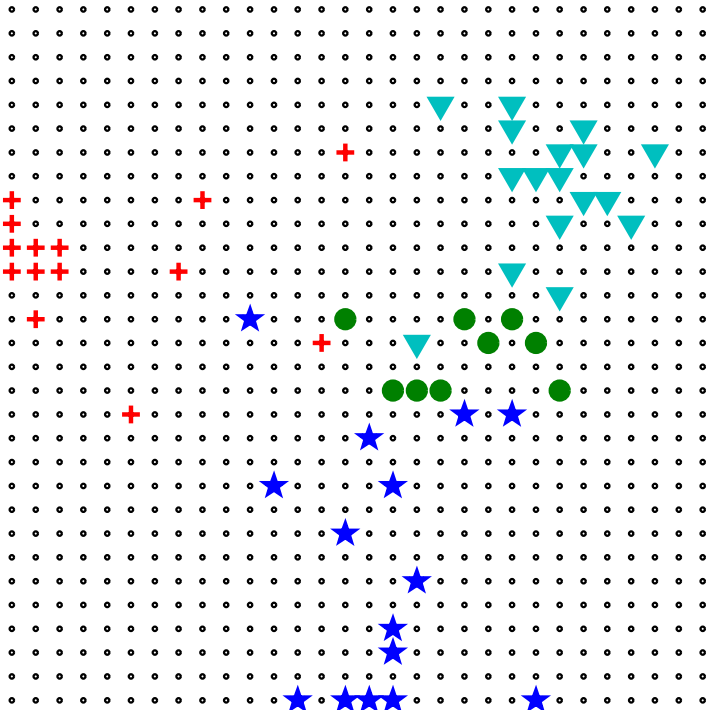
(a) 30-by-30 grid



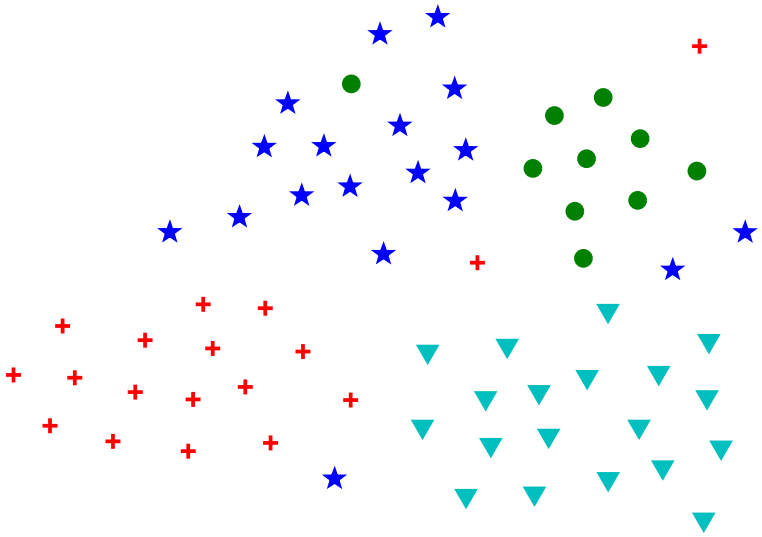
(b) 10-by-10 grid (Original)

(c) 3-by-3

Figure 4.8: The trained RGTM on the cat cortex data reused for new mappings in different grid sizes.



(a) Projection of data points to their position on a 30-by-30 grid by RGTM



(b) t-SNE projection

Figure 4.9: Visual comparison of the mapping of the cat cortex data, obtained by t-SNE and RGTM. Here, the RGTM was trained using a fine grid resolution, and only winner prototypes were labelled after training. Therefore, the ratio of labelled prototypes to original data points is approximately 1 to 1.18, so a one-to-one mapping is nearly achieved.

4.5 Summary

In this chapter, as noted in Table 4.4, we extended GTM towards data given by a dissimilarity matrix rather than Euclidean vectors. The resulting algorithm, relational GTM, can be used directly on the dissimilarity matrix. It has been demonstrated in the experiments that RGTM provides a reasonable topographic mapping of the data which is competitive to alternatives for clustering of dissimilarity data such as deterministic annealing or relational neural gas, and to alternatives for projection of dissimilarity data such as t-SNE and relational SOM.

Table 4.4: Extending GTM to relational data.

Topic	Relational	Out-of-Sample	Efficiency		Relevance
Technique	Data	Extension	vectorial	relational	Learning
GTM	√	√	√	?	√
t-SNE	√	?	?		?

Note that RGTM leads to a sparse representation of data in terms of a set of latent points in latent space together with a prescription of how this generates a probability distribution in data space. In particular, due to an explicit mapping of the latent space to the data space, an appropriate resolution of the mapping can be chosen posterior to training.

One drawback of RGTM compared to vectorial approaches is its dependency on the dissimilarity matrix which is quadratic in the number of points. This causes difficulties if large data sets are dealt with. In [59], approximation schemes have been proposed in the context of relational neural gas which, on the one hand, result in a sparse representation of prototypes, on the other hand, allow a patch processing of huge dissimilarity matrices for which the computational load would otherwise be too big. This way, the resulting topographic mapping scheme is linear in the number of data points. The transfer of this method to RGTM was presented in [179]. Another way to deal with large data sets is to incorporate the classical Nyström technique to approximate the dissimilarity matrix in linear time. This way, a linear time complexity algorithm is achieved as demonstrated in preliminary experiments in [47]. The comparison of both approximation techniques was discussed in [179]. It appears, that patch processing is well suited for streaming data, while Nyström approximation is good for dissimilarity matrices with low rank. An in-depth discussion on the Nyström technique as well as relational data in general will be given in the next chapter.

This chapter is based on: Andrej Gisbrecht and Frank-Michael Schleif. Metric and non-metric proximity transformations at linear costs. *Neurocomputing*. Submitted.

Chapter 5

Efficient processing of proximity data

In many application areas such as bioinformatics, text mining, image retrieval, spectroscopy domains or social networks the available electronic data are increasing and get more complex in size and representation. In general these data are not given in vectorial form and *domain specific* (dis-)similarity measures are used as a replacement or complement to Euclidean measures. These data are also often associated to dedicated structures which make a representation in terms of Euclidean vectors difficult: biological sequence data, text files, XML data, trees, graphs, or time series [25, 88, 115] are of this type. These data are inherently compositional and a feature representation leads to information loss. As an alternative, tailored dissimilarity measures such as pairwise alignment functions, kernels for structures or other domain specific similarity and dissimilarity functions can be used as the interface to the data. But also for vectorial data, non-metric proximity measures are common in some disciplines. An example of this type is the use of divergence measures [29] which are very popular for spectral data analysis in chemistry, geo- and medical sciences [114, 116, 149], and are not metric in general. In such cases, machine learning techniques which can deal with pairwise similarities or dissimilarities have to be used [119].

In previous chapter, we already discussed the relational GTM as a technique capable of dealing with such data. In section 4.3 we encountered issues which are also characteristic for most methods based on pairwise relations of data points. Typically, naive approaches have quadratic memory complexity and quadratic or even cubic runtime complexity. They are also often based on non-convex optimization schemes, which may result in convergence issues as shown in e.g. [67]. Thus, not only in dimensionality reduction, but also in other areas, such as e.g. clustering or classification, it is desirable to have fast and reliable (dis-)similarity based techniques. Kernel methods, readily available for a large variety of tasks, could solve this problem perfectly. They constitute some of the most effective and generic data mining techniques [140] and using the Nyström technique [173] they could be applied in an efficient way. Unfortunately, they are available only for metric similarities and can not be used for dissimilarities or non-metric similarities. Accordingly, it is of strong interest to get access to kernel approaches for a variety of potentially non-metric proximity data. In this chapter, we present a general approach which allows the transformation of dissimilarities to similarities and vice versa, but also from non-metric to metric data in linear time. Thus, it allows to apply relational and

kernel techniques on an arbitrary type of data in an efficient and consistent way.

The rest of the chapter is organized as follows. First we give a brief review of related work. Subsequently we review common transformation techniques for dissimilarity data and discuss the influence of non-Euclidean measures, by eigenvalue corrections. Thereafter we discuss alternative methods for processing small dissimilarity data. We extend this discussion to approximation strategies and give an alternative derivation of the Nyström approximation together with a convergence proof, also for indefinite kernels. This allows us to apply the Nyström technique to similarities as well as for dissimilarities. Thus, we can link both strategies effectively to use kernel methods for the analysis of (non-)metric dissimilarity data. Then we show the effectiveness of the proposed approach by different exemplary supervised experiments aligned with various error measures. We also discuss differences and similarities to some known approaches supported by experiments on simulated data.

5.1 Related work

Similarity and dissimilarity learning or for short proximity learning has attracted wide attention over the last years, pioneered by work of [52] and major contributions in [119] and different other research groups. As will be detailed more formally in the next section, the learning of proximities is challenging under different aspects: in general there is no underlying vector space, the proximities may be non-Euclidean, the data may not be metric. As mentioned before a matrix of metric similarities between objects is essentially a kernel and can be analysed by a multitude of kernel methods [140]. But complex preprocessing steps are necessary, as discussed in the following, to apply them on non-metric (dis-)similarities. Some recent work discussed non-metric *similarities* in the context of kernel approaches by means of indefinite kernels see e.g. [105, 122], resulting in non-convex formulations. Other approaches try to make the kernel representation positive semi definite (psd) [25, 27], but with high computational costs. In fact, as discussed in the work of Pekalska [119], non-Euclidean proximities can encode important information in the Euclidean as well as in non-Euclidean parts of space, represented by the positive and negative eigenvalues of the corresponding similarity matrix, respectively. Thus, transformations of similarities to make them psd, by e.g. truncating the negative eigenvalues, may be inappropriate [121]. This however is very data dependent and for a large number of datasets negative eigenvalues may actually stem from noise effects while for other data sets the negative eigenvalues carry relevant information [90, 91]. Often non-psd kernels are still used with kernel algorithms but actually on a heuristical basis, since corresponding error bounds are provided only for psd kernels in general. As we will see in the experiments for strongly non-psd data it may happen that standard kernel methods fail to converge due to the violation of underlying assumptions.

Another strategy tries to learn appropriate similarity functions based on the given data which then can be used for predictive models [3, 81]. A practical approach of the last type for classification problems was provided in [80]. The model is defined on a fixed set of landmarks per class and a transfer function, both heuristically optimized. The results are however in general substantially worse than those provided in [25] where the

datasets are taken from. The same authors extended this concept in [81] to the problem of regression incorporating a sparse optimization strategy on i.i.d. sampled landmarks. There the problem is formulated as a sparse linear regression problem. While very appealing the effectiveness of the approach for larger, realistic data sets including outliers is not addressed and it is not clear if the proposed approach is superior to other sparse regression approaches using strategies of [25, 26, 27] to address non-psd similarities.

In the following we will focus on non-metric proximities and especially *dissimilarities*. Native methods for the analysis of dissimilarity data have been proposed in [49, 55, 119], but are widely based on non-convex optimization schemes and with quadratic to linear memory and runtime complexity, the later employing some of the approximation techniques discussed subsequently and additional heuristics. The strategy to correct non-metric dissimilarities is addressed in the literature only for smaller data sets. This is caused by the two complicated steps of double centring and eigenvalue correction which are used in general and scale quadratic and cubic, respectively.

Large (dis-)similarity data are common in biology like the famous *UniProt-/SwissProt*-database with ≈ 500.000 entries or *GenBank* with ≈ 135.000 entries, but there are many more (dis-)similarity data as discussed in the work based on [119, 120]. These growing data sets request effective and generic modelling approaches, applicable for a wide range of data sets.

Here we will show how potentially non-metric (dis-)similarities can be effectively processed by standard kernel methods with linear costs, where linear can also apply to the transformation step. The proposed strategies permit the effective application of many kernel methods for these type of data under very mild conditions.

Especially for metric dissimilarities the approach keeps the known guarantees like generalization bounds (see e.g. [35]). For non-psd data we give a convergence proof, but the corresponding bounds are still open, yet our experiments are promising.

5.2 Transformation techniques for (dis-)similarities

Let $\mathbf{v}_j \in \mathbb{V}$ be a set of objects defined in some data space, with $|\mathbb{V}| = N$. We assume, there exists a dissimilarity measure such that $\mathbf{D} \in \mathbb{R}^{N \times N}$ is a dissimilarity matrix measuring the pairwise dissimilarities $D_{ij} = d(\mathbf{v}_j, \mathbf{v}_i)^2$ between all pairs $(\mathbf{v}_i, \mathbf{v}_j) \in \mathbb{V}$ [1]. Any reasonable (possibly non-metric) distance measure is sufficient. We assume zero diagonal $d(\mathbf{v}_i, \mathbf{v}_i) = 0$ for all i and symmetry $d(\mathbf{v}_i, \mathbf{v}_j) = d(\mathbf{v}_j, \mathbf{v}_i)$ for all i, j.

5.2.1 Transformation of dissimilarities and similarities into each other

Every dissimilarity matrix $\mathbf{D}$ can be seen as a distance matrix computed in some, not necessarily Euclidean, vector space. The matrix of the inner products computed in this space is the corresponding similarity matrix $\mathbf{S}$. It can be computed from $\mathbf{D}$ directly by a process referred to as double centring [119]:

$$\begin{aligned} \mathbf{S} &= -\mathbf{JDJ}/2 \\ \mathbf{J} &= (\mathbf{I} - \mathbf{1}\mathbf{1}^\top/N) \end{aligned}$$

[1] We assume D_{ij} to be squared to simplify the notation.

with identity matrix $\mathbf{I}$ and vector of ones $\mathbf{1}$. Similarly, it is possible to construct the dissimilarity matrix elementwise from the matrix of inner products $\mathbf{S}$

$$D_{ij} = S_{ii} + S_{jj} - 2S_{ij}.$$

As we can see, both matrices $\mathbf{D}$ and $\mathbf{S}$ are closely related to each other and represent the same data, up to translation, which is lost in the dissimilarity matrix.

The data stems from an Euclidean space, and therefore the distances d_{ij} are Euclidean, if and only if $\mathbf{S}$ is positive semi-definite (psd). This is the case, when we observe only non-negative eigenvalues in the eigenspectrum of the matrix $\mathbf{S}$ associated to $\mathbf{D}$. Such psd matrices $\mathbf{S}$ are also referred to as kernels and there are many classification techniques, which have been proposed to deal with such data, like the support vector machine (SVM). In the case of non-psd similarities, the kernel based techniques are no longer guaranteed to work properly and additional transformations of the data are required. To define these transformations we need first to understand the pseudo-Euclidean space.

5.2.2 Pseudo-Euclidean embedding

Given a symmetric dissimilarity with zero diagonal, an embedding of the data in a pseudo-Euclidean vector space is always possible [52].

Definition 1 (Pseudo-Euclidean space [119]) *A pseudo-Euclidean space $\xi = \mathbb{R}^{(p,q)}$ is a real vector space equipped with a non-degenerate, indefinite inner product $\langle .,. \rangle_\xi$. ξ admits a direct orthogonal decomposition $\xi = \xi_+ \oplus \xi_-$ where $\xi_+ = \mathbb{R}^p$ and $\xi_- = \mathbb{R}^q$ and the inner product is positive definite on ξ_+ and negative definite on ξ_-. The space ξ is therefore characterized by the signature (p, q).*

A symmetric bilinear form in this space is given by

$$\langle \mathbf{x}, \mathbf{y} \rangle_{p,q} = \sum_{i=1}^{p} x_i y_i - \sum_{i=p+1}^{p+q} x_i y_i = \mathbf{x}^\top \mathbf{I}_{p,q} \mathbf{y}$$

where $\mathbf{I}_{p,q}$ is a diagonal matrix with p entries 1 and q entries -1. Given the eigendecomposition of a similarity matrix $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ we can compute the corresponding vectorial representation $\mathbf{V}$ in the pseudo-Euclidean space by

$$\mathbf{V} = \mathbf{U}_{p+q} \left| \mathbf{\Lambda}_{p+q} \right|^{1/2} \tag{5.1}$$

where $\mathbf{\Lambda}_{p+q}$ consists of p positive and q negative non-zero eigenvalues and $\mathbf{U}_{p+q}$ consists of the corresponding eigenvectors. It is straightforward to see that $D_{ij} = \langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{v}_i - \mathbf{v}_j \rangle_{p,q}$ holds for every pair of data points.

Similarly to the signature (p, q) of a space ξ, we describe our finite data sets, given by a matrix $\mathbf{D}$ or $\mathbf{S}$, by the extended signature $(p, q, N - p - q)$ which represents the number of positive eigenvalues p, the number of negative eigenvalues q and the number of the remaining zero eigenvalues in the similarity matrix.

5.2.3 Dealing with pseudo-Euclidean data

In [25] different strategies were analysed to obtain valid kernel matrices for a given similarity matrix $\mathbf{S}$, most popular are: *flipping, clipping, vector-representation, shift correction*. The underlying idea is to remove negative eigenvalues in the eigenspectrum of the matrix $\mathbf{S}$. One may also try to learn an alternative psd kernel representation with maximum alignment to the original non-psd kernel matrix [25, 27, 103] or split the proximities based on positive and negative eigenvalues as discussed in [119, 122].

The *flip*-operation takes the absolute eigenvalues of the matrix $\mathbf{S}$. This corresponds to ignoring the separation of the space ξ into ξ_+ and ξ_- and instead computing in the space $\mathbb{R}^{p+q}$. This approach preserves the variation in the data and could be revoked for some techniques after the training by simply reintroducing the matrix $\mathbf{I}_{p,q}$ into the inner product.

The *shift*-operation increases all eigenvalues by the absolute value of the minimal eigenvalue. This approach performs a nonlinear transformation in the pseudo-Euclidean space, emphasizing ξ_+ and nearly eliminating ξ_-.

The *clip*-operation sets all negative eigenvalues to zero. This approach corresponds to ignoring the space ξ_- completely. As discussed in [121], depending on the data set, this space could carry important information and removing it would make some tasks, as e.g. classification, impossible.

After the transformation of the eigenvalues, the corrected matrix $\mathbf{S}^*$ is obtained as $\mathbf{S}^* = \mathbf{U}\mathbf{\Lambda}^*\mathbf{U}^\top$, with $\mathbf{\Lambda}^*$ as the modified eigenvalue matrix using one of the above operations. The obtained matrix $\mathbf{S}^*$ can now be considered as a valid kernel matrix $\mathbf{K}$ and kernel based approaches can be used to operate on the data.

The analysis in [121] indicates that for non-Euclidean dissimilarities some corrections like above may change the data representation such that information loss occurs. This however is not yet systematically explored and very data dependent, best supported by domain knowledge about the data or the used proximity measure.

Alternatively, techniques have been introduced which directly deal with possibly non-metric dissimilarities. Using the equation 5.1 the data can be embedded into the pseudo-Euclidean space. Classical vectorial machine learning algorithms can then be adapted to operate directly in the pseudo-Euclidean space. This can be achieved by e.g. defining a positive definite inner product in the space ξ. Variations of this approach are also possible whereby an explicit embedding is not necessary and the training can be done implicitly, based on the dissimilarity matrix only [119].

A further strategy is to employ the so called relational or proximity learning methods as we already discussed in chapter 4. More examples are presented in e.g. [49]. The underlying models consist of prototypes, which are implicitly defined as a weighted linear combination of training points:

$$\mathbf{w}_j = \sum_i \alpha_{ji}\mathbf{v}_i \text{ with } \sum_i \alpha_{ji} = 1\,.$$

But this explicit representation is not necessary because the algorithms are based only on a specific form of distance calculations using the matrix $\mathbf{D}$ and the potentially unknown vector space V is not needed. The basic idea is an implicit computation of distances

$d(\cdot,\cdot)$ during the model calculation based on the dissimilarity matrix $\mathbf{D}$ using weights α:

$$d(\mathbf{v}_i, \mathbf{w}_j)^2 = [\mathbf{D} \cdot \alpha_j]_i - \frac{1}{2} \cdot \alpha_j^\top \mathbf{D} \alpha_j. \tag{5.2}$$

Detailed explanation of this technique, on the example of relational GTM, can be seen in chapter 4. As shown e.g. in [59] the mentioned methods do not rely on a metric dissimilarity matrix $\mathbf{D}$, but it is sufficient to have a symmetric $\mathbf{D}$ in a pseudo-Euclidean space, with constant self-dissimilarities.

Dissimilarity space approach is another technique which does not embed the data into the pseudo-Euclidean space [119]. Instead, one selects a representative set of points $\mathbf{w}_i \in \mathbb{W}$ and considers for every point the dissimilarities to the set $\mathbb{W}$ as features, resulting in a vectorial representation $\mathbf{x}_i = [d(\mathbf{v}_i, \mathbf{w}_1), d(\mathbf{v}_i, \mathbf{w}_2), d(\mathbf{v}_i, \mathbf{w}_3), ...]^\top$. This corresponds to an embedding into an Euclidean space with the dimensionality equal to the size of the selected set of points. These vectors can then be processed using any vectorial approaches. A negative point of this representation is the change of the original data representation which may disturb the structure of the data. It is also highly reliable on a good representative set, since highly correlated sampled points generate similar features and the correlation information is lost in the embedded space.

5.2.4 Complexity

The methods discussed before are suitable for data analysis based on similarity or dissimilarity data where the number of samples N is rather small, e.g. scales by some thousand samples. For large N, most of the techniques discussed above become infeasible. All techniques which use the full (dis-)similarity matrix, have $\mathcal{O}(N^2)$ memory complexity and thus at least $\mathcal{O}(N^2)$ computational complexity.

Double centring, if done naively, is cubic, although after simplifications it can be computed in $\mathcal{O}(N^2)$. Transformation from $\mathbf{S}$ to $\mathbf{D}$ can be done elementwise, but if the full matrix is required it is still quadratic.

All the techniques relying on the full eigenvalue decomposition, e.g. for eigenvalue correction or for explicit pseudo-Euclidean embedding, have an $\mathcal{O}(N^3)$ computational complexity. Relational GTM and other methods working implicitly by using the dissimilarity matrix have at least quadratic complexity.

The only exception is the dissimilarity space approach. If it possible to select a good representative set of a small size, one can achieve linear computational and memory complexity. The technique becomes quadratic as well, if all data points are selected for representative set.

Other then this, only for *metric, similarity data* (psd kernels) efficient approaches have been proposed before, e.g. the Core-Vector Machine (CVM) [154] or low-rank linearised SVM [178] for classification problems or an approximated kernel k-means algorithm for clustering [28].

A schematic view of the relations between $\mathbf{S}$ and $\mathbf{D}$ and its transformations is shown in Figure 5.1, including the complexity of the transformations. Some of the steps can be done more efficiently by known methods, but with additional constraints or in atypical settings as discussed in the following.

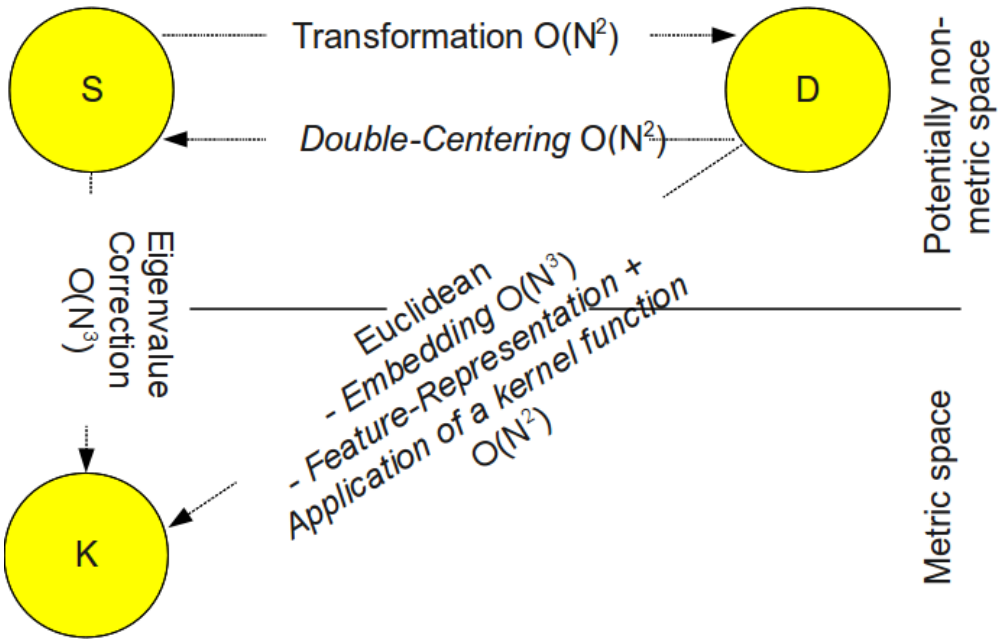


Figure 5.1: Schema of the relation between similarities and dissimilarities.

In the following, we discuss techniques to deal with larger sample sets for potentially non-metric similarity and especially dissimilarity data. We show how standard kernel methods can be used, assuming that for non-metric data, the necessary transformations have no severe negative influence on the data accuracy. Basically also core-set techniques [2] become accessible for large potentially non-metric (dis-)similarity data in this way, but at the cost of multiple additional intermediate steps. In particular, we investigate the Nyström approximation technique, as low rank linear time approximation technique; we will show its suitability and linear time complexity for similarities as well as dissimilarities, applied on the raw data as well as for the eigenvalue correction.

5.3 Nyström approximation

As shown in [173], given a symmetric positive semi-definite kernel matrix $\mathbf{K}$, it is possible to create a low rank approximation of this matrix using the Nyström technique [117]. The idea is to sample m points, the so called landmarks, and to analyse the small $m \times m$ kernel matrix $\mathbf{K}_{m,m}$ constructed from the landmarks. The eigenvalues and eigenvectors from the matrix $\mathbf{K}_{m,m}$ can be used to approximate the eigenvalues and eigenvectors of the original matrix $\mathbf{K}$. This allows to represent the complete matrix in terms of a linear part of the full matrix only. The final approximation takes the simple form

$$\hat{\mathbf{K}} = \mathbf{K}_{N,m}\mathbf{K}_{m,m}^{-1}\mathbf{K}_{m,N}, \tag{5.3}$$

where $\mathbf{K}_{N,m}$ is the kernel matrix between N data points and m landmarks and $\mathbf{K}_{m,m}^{-1}$ is the Moore-Penrose pseudoinverse of the small matrix.

This technique has been proposed in the context of Mercer kernel methods in [173] with related proofs and bounds given in [35] and very recent results in [50]. It can be applied in conjunction with algorithms using the kernel matrix in multiplications with other matrices or vectors only. Due to the explicit low rank form as in Equation (5.3) it is possible to select the order of multiplication, thus reducing the complexity from quadratic in the number of data points to a linear one.

5.3.1 Eigenvalue decomposition of a Nyström approximated matrix

In some applications it might be useful to compute the exact eigenvalue decomposition of the approximated matrix $\hat{\mathbf{K}}$, e.g. to compute the pseudo-inverse of this matrix. We will show now, how this decomposition can be computed in linear time. The psd matrix approximated by Equation (5.3) can be written as

$$\begin{aligned}\hat{\mathbf{K}} &= \mathbf{K}_{N,m}\mathbf{K}_{m,m}^{-1}\mathbf{K}_{m,N} \\ &= \mathbf{K}_{N,m}\mathbf{U}\mathbf{\Lambda}^{-1}\mathbf{U}^\top\mathbf{K}_{N,m}^\top \\ &= \mathbf{B}\mathbf{B}^\top,\end{aligned}$$

where we defined $\mathbf{B} = \mathbf{K}_{N,m}\mathbf{U}\mathbf{\Lambda}^{-1/2}$ with $\mathbf{U}$ and $\mathbf{\Lambda}$ being the eigenvectors and eigenvalues of $\mathbf{K}_{m,m}$, respectively. Further it follows

$$\begin{aligned}\hat{\mathbf{K}}^2 &= \mathbf{B}\mathbf{B}^\top\mathbf{B}\mathbf{B}^\top \\ &= \mathbf{B}\mathbf{V}\mathbf{A}\mathbf{V}^\top\mathbf{B}^\top,\end{aligned}$$

where $\mathbf{V}$ are the orthonormal eigenvectors of the matrix $\mathbf{B}^\top\mathbf{B}$ and $\mathbf{A}$ the matrix of its eigenvalues. The corresponding eigenequation can be written as $\mathbf{B}^\top\mathbf{B}\mathbf{v} = a\mathbf{v}$. Multiplying it with $\mathbf{B}$ from left we get the eigenequation for $\hat{\mathbf{K}}$

$$\mathbf{B}\mathbf{B}^\top(\mathbf{B}\mathbf{v}) = a\,(\mathbf{B}\mathbf{v})\,.$$

It is clear, that $\mathbf{A}$ must be the matrix of eigenvalues of $\hat{\mathbf{K}}$. The matrix $\mathbf{B}\mathbf{v}$ is the matrix of the corresponding eigenvectors, which are orthogonal but not necessary orthonormal. The normalization can be computed from the decomposition

$$\begin{aligned}\hat{\mathbf{K}} &= \mathbf{B}\mathbf{V}\mathbf{V}^\top\mathbf{B}^\top \\ &= \mathbf{B}\mathbf{V}\mathbf{A}^{-1/2}\mathbf{A}\mathbf{A}^{-1/2}\mathbf{V}^\top\mathbf{B}^\top \\ &= \mathbf{C}\mathbf{A}\mathbf{C}^\top,\end{aligned}$$

where we defined $\mathbf{C} = \mathbf{B}\mathbf{V}\mathbf{A}^{-1/2}$ as the matrix of orthonormal eigenvectors of $\hat{\mathbf{K}}$. Thus, $\hat{\mathbf{K}} = \mathbf{C}\mathbf{A}\mathbf{C}^\top$ is the orthonormal eigendecomposition of $\hat{\mathbf{K}}$.

5.3.2 Convergence proof

The Nyström approximation was proposed for the psd matrices and thus, it was not accessible for distance matrices and similarities coming from non-psd kernel functions. First developments to apply the Nyström technique to indefinite matrices were presented in [47, 49]. Although supported with experiments, a formal proof was lacking. Here we present a proof, that in fact, if the number of landmarks is large enough, the Nyström approximated, possible indefinite kernel converges in the operator norm to the true underlying kernel. Generalization bounds will be a subject of the future work.

Let K be an integral operator and its kernel $k \in L^2(\Omega^2)$ be a continuous symmetric function (not necessarily psd):

$$Kf(x) := \int_\Omega k(x,y)f(y)d\mu(y).$$

Without loss of generality let Ω be an interval $[a,b] \subset \mathbb{R}$ with measure 1. Then K is a compact operator in a Hilbert space $\mathfrak{H}$

$$\|K\|_{L^2 \to L^2} := \sup_x \int_\Omega |k(x,y)|d\mu(y) \leq \|k\|_\infty,$$

with the operator norm $\|.\|_{L^2 \to L^2}$ and supremum norm $\|.\|_\infty$.

We define a measurement operator T_m which divides the space Ω into m spaces Ω_j, each with the measure $1/m$. It converts functions $f \in \mathfrak{H}$ to functions $f_m \in \mathfrak{H}_m$ which are piecewise constant on each Ω_j. The corresponding integral kernel of T_m is defined as:

$$t_m(x,y) := \begin{cases} m & x,y \in \Omega_j \text{ for any } j \\ 0 & \text{else.} \end{cases}$$

It follows for an $x \in \Omega_j$ that

$$T_m f(x) = \int_\Omega t_m(x,y)f(y)d\mu(y) = m\int_{\Omega_j} f(y)d\mu(y),$$

where we can see, that the right hand side is the mean value of $f(y)$ on Ω_j and thus constant for all $x \in \Omega_j$. This way, the operator T_m allows us to approximate a function $f(x)$ by measuring it at m places $f(x_j)$ and assuming that it is constant in between. Measuring the operator K we get $K_m := T_m \circ K$ with the integral kernel

$$\begin{aligned} \int_\Omega t_m(x,z)k(z,y)d\mu(z) &= \sum_{j=1}^m \int_{\Omega_j} t_m(x,z)k(z,y)d\mu(z) \\ &= \sum_{j=1}^m 1_{\Omega_j}(x)m\int_{\Omega_j} k(z,y)d\mu(z) \\ &= \sum_{j=1}^m 1_{\Omega_j}(x)k_j(y) \\ &=: k_m(x,y), \end{aligned}$$

where $1_{\Omega_j}(x)$ is the indicator function which is 1 if $x \in \Omega_j$ and 0 elsewhere and we defined $k_j = m \int_{\Omega_j} k(z,y)d\mu(z)$.

We can now analyse the convergence behaviour of K_m to K. $\forall x \in \Omega_j$ and $\forall y \in \Omega$ we get

$$\begin{aligned}
&|k_m(x,y) - k(x,y)| = \\
&= \left| m \int_{\Omega_j} k(z,y)d\mu(z) - m \int_{\Omega_j} k(x,y)d\mu(z) \right| \\
&\leq m \int_{\Omega_j} |k(z,y) - k(x,y)| \, d\mu(z).
\end{aligned}$$

Since k is continuous on the interval $[a,b]$, it is uniformly continuous and we can bound

$$\begin{aligned}
|k(z,y) - k(x,y)| \leq \mathcal{D}(\Omega_j) &:= \sup_{\substack{x_1, x_2 \in \Omega_j \\ y \in \Omega}} |k(x_1,y) - k(x_2,y)| \\
&\leq \delta_m := \max_j \mathcal{D}(\Omega_j)
\end{aligned}$$

and therefore

$$\sup_{\substack{x \in \Omega \\ y \in \Omega}} |k_m(x,y) - k(x,y)| \leq \delta_m.$$

For $m \to \infty$ the Ω_j become smaller and $\delta_m \to 0$, thus kernel k_m converges to k. For the operators K and K_m it follows

$$\|K_m - K\|_{L^2 \to L^2} \to 0$$

which shows that K_m converges to K in the operator norm, if the number of measurements goes to infinity.

Applying K_m on f results in

$$\begin{aligned}
K_m f(x) &= \int_\Omega k_m(x,y) f(y) d\mu(y) \\
&= \sum_{j=1}^{m} 1_{\Omega_j}(x) \int_\Omega k_j(y) f(y) d\mu(y) \\
&= \sum_{j=1}^{m} a_j 1_{\Omega_j}(x)
\end{aligned}$$

where $a_j := \int_\Omega k_j(y) f(y) d\mu(y)$ is a constant with respect to x. It is clear that $K_m f$ is always in the linear hull of $1_{\Omega_1}(x), ..., 1_{\Omega_m}(x)$ and the image of the operator $\Im K_m = \text{span}\{1_{\Omega_1}(x), ..., 1_{\Omega_m}(x)\}$ is m dimensional. Since the coefficients a_j are finite, K_m is a compact operator and because the sequence of K_m converges to K, we see that K is in fact a compact operator.

According to the "Perturbation of bounded operators" theorem [166], if a sequence K_m converges to K in the operator norm, then for an isolated eigenvalue λ of K there

exist isolated eigenvalues λ_m of K_m such that $\lambda_m \to \lambda$ and the corresponding spectral projections converge in operator norm. This theorem allows us to estimate the eigenvalues and eigenfunctions of the unknown operator K by computing the eigendecomposition of the measured operator K_m.

The eigenfunctions and eigenvalues of the operator K_m are given as the solutions of the eigenequation

$$K_m f = \lambda f. \tag{5.4}$$

We know that the left hand side of the equation is in the image of K_m and therefore an eigenfunction f must have the form

$$f(x) = \sum_{i=1}^{m} f_i 1_{\Omega_i}(x) \tag{5.5}$$

where f_i are constants. For the left side of the Equation (5.4) it follows

$$\begin{aligned} K_m f(x) &= \int_\Omega \sum_{j=1}^{m} 1_{\Omega_j}(x) k_j(y) f(y) d\mu(y) \\ &= \sum_{j=1}^{m} 1_{\Omega_j}(x) \int_\Omega k_j(y) \sum_{i=1}^{m} f_i 1_{\Omega_i}(y) d\mu(y) \\ &= \sum_{j=1}^{m} \sum_{i=1}^{m} 1_{\Omega_j}(x) f_i \int_{\Omega_i} k_j(y) d\mu(y) \\ &= \sum_{j=1}^{m} \sum_{i=1}^{m} 1_{\Omega_j}(x) \frac{1}{m} f_i k_{ji} \end{aligned}$$

and we defined $k_{ji} = m \int_{\Omega_i} k_j(y) d\mu(y) = m^2 \int_{\Omega_i} \int_{\Omega_j} k(y,z) d\mu(y) d\mu(z)$ which represents our measurement of the kernel k around the i-th and j-th points. If we combine the above equation with the Equation (5.4) for an $x \in \Omega_j$ we get

$$\sum_{i=1}^{m} \frac{1}{m} k_{ji} f_i = \lambda f_j.$$

This equation is a weighted eigenequation and we can turn it into a regular eigenequation by defining $\tilde{\lambda} = m\lambda$ and $\tilde{f}_i = f_i/\sqrt{m}$. Thus, we get

$$\sum_{i=1}^{m} k_{ji} \tilde{f}_i = \tilde{\lambda} \tilde{f}_j.$$

Hence $\tilde{\lambda}$ and $\tilde{f}$ are the eigenvalues and eigenvectors of matrix (k_{ji}). Note, that f_i are

scaled to guarantee the normalization of $\tilde{f}$

$$\begin{aligned} 1 &= \int_\Omega f(x)f(x)d\mu(x) \\ &= \int_\Omega \sum_{i=1}^{m} f_i^2 1_{\Omega_i}(x)d\mu(x) \\ &= \sum_{i=1}^{m} f_i^2 \int_{\Omega_i} d\mu(x) \\ &= \sum_{i=1}^{m} \left(\frac{f_i}{\sqrt{m}}\right)^2. \end{aligned}$$

The eigendecomposition takes the form

$$(k_{ji}) = \sum_{l=1}^{m} \tilde{\lambda}^l \tilde{f}^l (\tilde{f}^l)'$$

and for a single measured element we get

$$k_{ij} = \sum_{l=1}^{m} \tilde{\lambda}^l \tilde{f}_i^l \tilde{f}_j^l.$$

According to the spectral theorem [172] the eigendecomposition of k is

$$k(x,y) = \sum_{l=1}^{\infty} \gamma^l \phi^l(x)\phi^l(y)$$

where γ^l and ϕ^l are the eigenvalues and eigenfunctions, respectively. Since K is a compact operator, γ^l is a null sequence. Thus, the sequence of operators $\tilde{K}_m$ with the kernel $\tilde{k}_m(x,y) = \sum_{l=1}^{m} \gamma^l \phi^l(x)\phi^l(y)$ converges to K in the operator norm for $m \to \infty$ [172] and we can approximate

$$\begin{aligned} k(x,y) &\approx \sum_{l=1}^{m} \gamma^l \phi^l(x)\phi^l(y) \\ &= \sum_{l=1}^{m} \int_\Omega k(x,z)\phi^l(z)d\mu(z)\frac{1}{\gamma^l}\int_\Omega k(y,z')\phi^l(z')d\mu(z'), \end{aligned}$$

where we assume that none of the γ^l are zero. Further, due to the "Perturbation of bounded operators" theorem, the eigenvalues λ^l converge to γ^l and the corresponding eingenspaces converge in the operator norm and we can approximate

$$k(x,y) \approx \sum_{l=1}^{m} \int_\Omega k(x,z)f^l(z)d\mu(z)\frac{1}{\lambda^l}\int_\Omega k(y,z')f^l(z')d\mu(z').$$

Taking into account the Equation (5.5) the above formula turns into

$$\begin{aligned}
k(x,y) \approx & \sum_{l=1}^{m} \int_{\Omega} k(x,z) \sum_{i=1}^{m} f_i^l 1_{\Omega_i}(z) d\mu(z) \\
& \cdot \frac{1}{\lambda^l} \int_{\Omega} k(y,z') \sum_{j=1}^{m} f_j^l 1_{\Omega_j}(z') d\mu(z') \\
= & \sum_{l=1}^{m} \sum_{i=1}^{m} f_i^l \int_{\Omega_i} k(x,z) d\mu(z) \frac{1}{\lambda^l} \sum_{j=1}^{m} f_j^l \int_{\Omega_j} k(y,z') d\mu(z') \\
= & \sum_{i=1}^{m} \sum_{j=1}^{m} k_i(x) \left(\sum_{l=1}^{m} \frac{f_i^l}{\sqrt{m}} \frac{1}{m\lambda^l} \frac{f_j^l}{\sqrt{m}} \right) k_j(y) \\
= & \sum_{i=1}^{m} \sum_{j=1}^{m} k_i(x) \left(k^{-1}\right)_{ij} k_j(y),
\end{aligned}$$

where k^{-1} is the pseudo-inverse of the matrix consisting of elements k_{ij}. It is now clear, that after measuring $k_i(x)$ at N places and writing the above formula in matrix form, we retain the original Nyström approximation as in Equation (5.3).

Note, that the approximation of $k(x, y)$ consists of two approximations. The first one is the approximation of the rank of the matrix and the second one is the approximation of the eigenfunctions and eigenvalues. Although we don't know the exact eigenvalues and eigenfunctions of kernel $k(x, y)$, the approximation is exact if the kernel has the rank m. This fact is known for the Nyström approximation and can be validated by simple matrix transformations. The reason is, that if the rank of a kernel is m then it can be represented as an inner product in a pseudo-Euclidean space and m linearly independent landmarks build a basis which spans this space. The position of any new point x is then fully determined by $k(x, x_i)$, with x_i being the landmarks, so that all inner products between any points are determined and the matrix $\mathbf{K}$ can be computed precisely.

The Nyström approximation involves the computation of $\mathbf{K}_{N,m}$ and inversion of $\mathbf{K}_{m,m}$ with the corresponding complexities of $\mathcal{O}(mN)$ and $\mathcal{O}(m^3)$, respectively. The multiplication of both matrices as well as multiplication of the approximated matrix with other matrices, required for further processing and training, has the complexity of $\mathcal{O}(m^2N)$. Thus, the overall complexity of the Nyström technique is given by $\mathcal{O}(m^2N)$.

5.4 Transformations of (dis-)similarities with linear costs

The Nyström approximation was proposed originally to deal with large psd similarity matrices with kernel approaches in mind [173]. To apply these techniques on indefinite similarity and dissimilarity matrices additional transformations, as discussed in section 5.2, are required. Unfortunately, these transformations have quadratic or even cubic time complexity, making the advantage gained by the Nyström approximation pointless. Since we can now apply the Nyström technique on arbitrary symmetric matrices, it is not only possible to approximate the dissimilarities directly, but also to perform the

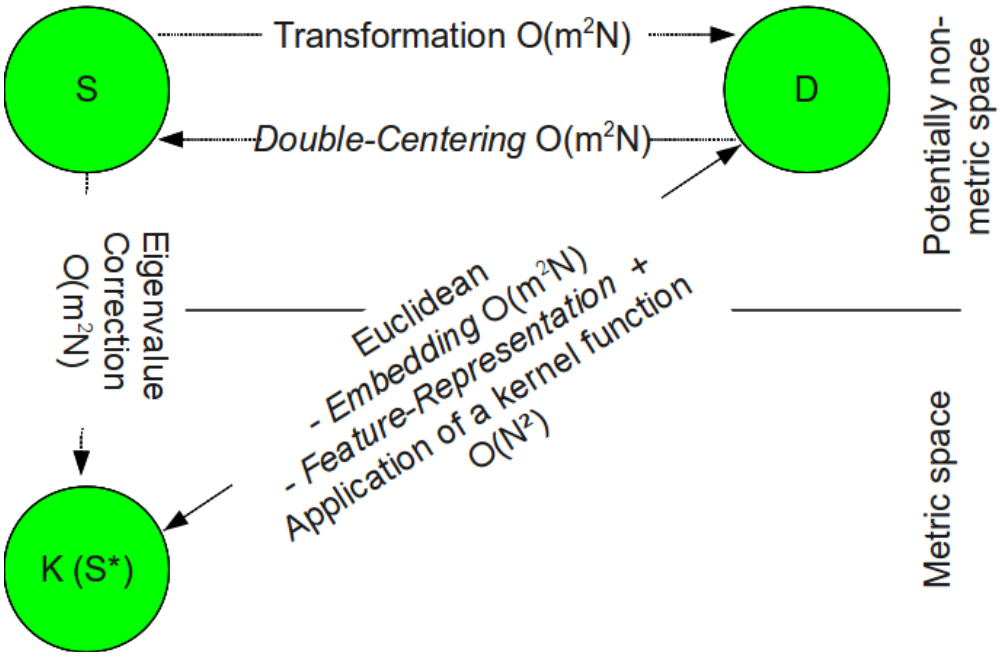


Figure 5.2: Updated schema from Figure 5.1 using the discussed approximation. The costs are now substantially smaller, provided $m \ll N$.

transformations in linear time. Thus, we can apply relational and kernel techniques on similarities and dissimilarities regardless, performing eigenvalue correction if necessary.

In this section we will elaborate how the transformations discussed in section 5.2 can be done in linear time if applied for the Nyström-approximated matrices. The updated costs are shown on the Figure 5.2.

5.4.1 Transformation of dissimilarities and similarities into each other

Given a dissimilarity matrix $\mathbf{D}$, there are two ways to construct the approximated matrix $\hat{\mathbf{S}}$. First, we can transform $\mathbf{D}$ to $\mathbf{S}$ using double centring and then apply Nyström approximation to $\mathbf{S}$. Obviously, this approach has quadratic time complexity due to the double centring step. Second, we can approximate $\mathbf{D}$ to $\hat{\mathbf{D}}$ first and then apply double centring. As we will show in the following, this transformation requires only linear computational time.

As mentioned before, from the dissimilarity matrix $\mathbf{D}$ we can compute the corresponding similarity matrix using double centring. This process is noted as $\mathbf{S}(\mathbf{D})$ in the following:

$$\mathbf{S}(\mathbf{D}) = -\mathbf{JDJ}/2$$

where $\mathbf{J} = (\mathbf{I} - \mathbf{1}\mathbf{1}^\top/N)$ with identity matrix $\mathbf{I}$ and vector of ones $\mathbf{1}$. Expanding the right

side of the equation we get

$$\begin{aligned}\mathbf{S}(\mathbf{D}) &= -\frac{1}{2}\mathbf{JDJ} \\ &= -\frac{1}{2}\left(\left(\mathbf{I}-\frac{1}{N}\mathbf{1}\mathbf{1}^\top\right)\mathbf{D}\left(\mathbf{I}-\frac{1}{N}\mathbf{1}\mathbf{1}^\top\right)\right) \\ &= -\frac{1}{2}\left(\mathbf{D}-\frac{1}{N}\mathbf{D}\mathbf{1}\mathbf{1}^\top-\frac{1}{N}\mathbf{1}\mathbf{1}^\top\mathbf{D}+\frac{1}{N^2}\mathbf{1}\mathbf{1}^\top\mathbf{D}\mathbf{1}\mathbf{1}^\top\right).\end{aligned}$$

Approximating $\mathbf{S}(\mathbf{D})$ requires computation of a linear part of each summand, but still involves summation over the full matrix $\mathbf{D}$.

Alternatively, by approximating $\mathbf{D}$ first, we get

$$\begin{aligned}\mathbf{S} \overset{Ny}{\approx}\ \mathbf{S}(\hat{\mathbf{D}}) = -\frac{1}{2}\Big[&\mathbf{D}_{N,m}\cdot\mathbf{D}_{m,m}^{-1}\cdot\mathbf{D}_{m,N}-\frac{1}{N}\mathbf{D}_{N,m} \\ &\cdot(\mathbf{D}_{m,m}^{-1}\cdot(\mathbf{D}_{m,N}\mathbf{1}))\mathbf{1}^\top-\frac{1}{N}\mathbf{1}((\mathbf{1}^\top\mathbf{D}_{N,m})\cdot\mathbf{D}_{m,m}^{-1}) \\ &\cdot\mathbf{D}_{m,N}+\frac{1}{N^2}\mathbf{1}((\mathbf{1}^\top\mathbf{D}_{N,m})\cdot\mathbf{D}_{m,m}^{-1}\cdot(\mathbf{D}_{m,N}\mathbf{1}))\mathbf{1}^\top\Big].\end{aligned} \tag{5.6}$$

This equation can be rewritten for each entry of the matrix $\mathbf{S}(\hat{\mathbf{D}})$

$$\begin{aligned}S_{ij}(\hat{\mathbf{D}}) = -\frac{1}{2}\Big[&\mathbf{D}_{i,m}\cdot\mathbf{D}_{m,m}^{-1}\cdot\mathbf{D}_{m,j} \\ &-\frac{1}{N}\sum_k\mathbf{D}_{k,m}\cdot\mathbf{D}_{m,m}^{-1}\cdot\mathbf{D}_{m,j} \\ &-\frac{1}{N}\sum_k\mathbf{D}_{i,m}\cdot\mathbf{D}_{m,m}^{-1}\cdot\mathbf{D}_{m,k} \\ &+\frac{1}{N^2}\sum_{kl}\mathbf{D}_{k,m}\cdot\mathbf{D}_{m,m}^{-1}\cdot\mathbf{D}_{m,l}\Big],\end{aligned}$$

as well as for the sub-matrices $\mathbf{S}_{m,m}(\hat{\mathbf{D}})$ and $\mathbf{S}_{N,m}(\hat{\mathbf{D}})$, in which we are interested for the Nyström approximation

$$\begin{aligned}\mathbf{S}_{m,m}(\hat{\mathbf{D}}) = -\frac{1}{2}\Big[&\mathbf{D}_{m,m}-\frac{1}{N}\mathbf{1}\cdot\sum_k\mathbf{D}_{k,m} \\ &-\frac{1}{N}\sum_k\mathbf{D}_{m,k}\cdot\mathbf{1}^\top \\ &+\frac{1}{N^2}\mathbf{1}\cdot\sum_{kl}\mathbf{D}_{k,m}\cdot\mathbf{D}_{m,m}^{-1}\cdot\mathbf{D}_{m,l}\cdot\mathbf{1}^\top\Big]\end{aligned}$$

$$\begin{aligned}\mathbf{S}_{N,m}(\hat{\mathbf{D}}) &= -\frac{1}{2}\Bigg[\mathbf{D}_{N,m} - \frac{1}{N}\mathbf{1}\cdot\sum_k \mathbf{D}_{k,m} \\ &\quad -\frac{1}{N}\sum_k \mathbf{D}_{N,m}\cdot\mathbf{D}_{m,m}^{-1}\cdot\mathbf{D}_{m,k}\cdot\mathbf{1}^\top \\ &\quad +\frac{1}{N^2}\mathbf{1}\cdot\sum_{kl}\mathbf{D}_{k,m}\cdot\mathbf{D}_{m,m}^{-1}\cdot\mathbf{D}_{m,l}\cdot\mathbf{1}^\top\Bigg].\end{aligned}$$

Now, the matrix $\mathbf{S}(\hat{\mathbf{D}})$ can be approximated via the matrix $\hat{\mathbf{S}}(\hat{\mathbf{D}})$ using the matrices $\mathbf{S}_{m,m}(\hat{\mathbf{D}})$ and $\mathbf{S}_{N,m}(\hat{\mathbf{D}})$. This requires only a linear part of $\mathbf{D}$ and involves linear computation time.

Comparing this approach to the quadratic computation of $\mathbf{S}_{N,m}$, we see, that the first three summands are identical and only the forth summand is different. This term involves summation over the full dissimilarity matrix and, depending on the approximation quality of $\hat{\mathbf{D}}$, might vary. The deviation is added to each pairwise similarity resulting in a nonlinear transformation of the data. If m corresponds to the rank of $\mathbf{D}$ then double centring is exact and no information loss occurs during the approximation. Otherwise, the information loss increases with smaller m for both approaches and the error is made by approximating $\mathbf{S}$ in the first case and by approximating $\mathbf{D}$ in the second case. If the Nyström approximation is feasible for a given data set, then the second approach allows to perform the transformation in linear instead of quadratic time.

It should be mentioned that a similar transformation is possible with the landmark multidimensional scaling (LMDS) [33]. The idea is to sample a small amount m of points, the so called landmarks, compute the corresponding dissimilarity matrix followed by a double centring on this matrix. Finally the data are projected to a low dimensional space using an eigenvalue decomposition. The remaining points can then be projected into the same space, taking into account the distances to the landmarks, and applying a triangulation. From this vectorial representation of the data one can easily retrieve the similarity matrix as a scalar product between the points.

It was shown, that LMDS is a Nyström technique as well [126], but compared to our proposed approach in Equation (5.6) it makes not only an error in the forth summand, but also in the second and the third. Additionally, and more importantly, by projecting into Euclidean space it makes an implicit clipping of the eigenvalues. As discussed above and will be shown later, this might disturb data significantly, leading to qualitatively worse results. Thus, our proposed method can be seen as a generalization of LMDS and should be used instead.

Similarly to the transformation from $\mathbf{D}$ to $\hat{\mathbf{S}}$, there are two ways to transform $\mathbf{S}$ to $\hat{\mathbf{D}}$. First, transform the full matrix $\mathbf{S}$ to $\mathbf{D}$ using $D_{ij} = S_{ii} + S_{jj} - 2S_{ij}$ and then apply the Nyström approximation

$$\hat{\mathbf{D}} = \mathbf{D}_{N,m}\mathbf{D}_{m,m}^{-1}\mathbf{D}_{N,m}^\top. \tag{5.7}$$

Second, approximate $\mathbf{S}$ with $\hat{\mathbf{S}}$ and then transform it to $\hat{\mathbf{D}}$. The first approach requires quadratic time, since it transforms the full matrix. In the second approach only $\mathbf{D}_{N,m}$ is computed, thus making it linear in time and memory. Obviously, both approaches produce the same results, but the second one is significantly faster. The reason is, that

for the computation of $\hat{\mathbf{D}}$ only the matrix $\mathbf{D}_{N,m}$ is required and it is not necessary to compute the rest of $\mathbf{D}$.

5.4.2 Eigenvalue correction

For non-Euclidean data, the corresponding similarity matrix is indefinite. We would like to make the data Euclidean in order to avoid convergence issues, or to be able to use kernel methods. A strategy to obtain a valid kernel matrix from similarities is to apply an eigenvalue correction as discussed in section 5.2.3. This however can be prohibitive for large matrices, since to correct the whole eigenvalue spectrum, the whole eigenvalue decomposition is needed, which has $\mathcal{O}(N^3)$ complexity. The Nyström approximation can again decrease computational costs dramatically. Since we can now apply the approximation on an arbitrary symmetric matrix, we can make the correction afterwards, reducing the complexity to a linear one, as we will show now.

Given non-metric dissimilarities $\mathbf{D}$, we can first approximate them and then convert to approximated similarities $\hat{\mathbf{S}}(\hat{\mathbf{D}})$ using the Equation (5.6). For similarities $\hat{\mathbf{S}}$, given directly or obtained from $\hat{\mathbf{S}}(\hat{\mathbf{D}})$, we need to compute the eigenvalue decomposition in linear time. As we have shown in the section 5.3.1, it is possible to compute the exact eigenvalue decomposition of a Nyström-approximated psd matrix in linear time. Since $\hat{\mathbf{S}}$ is indefinite, we can not apply the above technique directly. Instead, since in a squared matrix the eigenvectors stay the same, we first compute

$$\begin{aligned}\hat{\mathbf{S}}^2 &= \mathbf{S}_{N,m}\mathbf{S}_{m,m}^{-1}\left(\mathbf{S}_{m,N}\cdot\mathbf{S}_{N,m}\right)\mathbf{S}_{m,m}^{-1}\mathbf{S}_{m,N}\\ &= \mathbf{S}_{N,m}\tilde{\mathbf{S}}_{m,m}\mathbf{S}_{N,m}^{\top}.\end{aligned}$$

The resulting matrix can be computed in linear time and is psd. This means, we can determine its eigenvalue decomposition as described in section 5.3.1:

$$\hat{\mathbf{S}}^2 = \mathbf{C}\tilde{\mathbf{A}}\mathbf{C}^{\top},$$

where $\tilde{\mathbf{A}}$ are the eigenvalues of $\hat{\mathbf{S}}^2$ and $\mathbf{C}$ are the eigenvectors of both $\hat{\mathbf{S}}^2$ and $\hat{\mathbf{S}}$.

Using the eigenvectors $\mathbf{C}$, the eigenvalues $\mathbf{A}$ of $\hat{\mathbf{S}} = \mathbf{C}\mathbf{A}\mathbf{C}^{\top}$ can be retrieved via $\mathbf{A} = \mathbf{C}^{\top}\hat{\mathbf{S}}\mathbf{C}$. Then we can correct the eigenvalues $\mathbf{A}$ by some technique as discussed in section 5.2.3 to $\mathbf{A}^*$. The corrected approximated matrix $\hat{\mathbf{S}}^*$ is then simply

$$\hat{\mathbf{S}}^* = \mathbf{C}\mathbf{A}^*\mathbf{C}^{\top}. \tag{5.8}$$

Thus, using a low rank representation of a similarity matrix we can compute its eigenvalue decomposition and perform eigenvalue correction in linear time. If it is desirable to work with the corrected dissimilarities, then using the Equation (5.7), it is possible to transform the corrected similarity matrix $\hat{\mathbf{S}}^*$ back to dissimilarities resulting in the corrected and approximated matrix $\hat{\mathbf{D}}^*$.

5.4.3 Out-of-sample extension

Usually models are learned by a training set and we expect them to generalize well on the new unseen data, or the test set. In such cases we need to provide an out-of-sample

extension, i.e. a way to apply the model on the new data. This might be a problem for the techniques dealing with (dis-)similarities. If the matrices are corrected, we need to correct the new (dis-)similarities as well to get consistent results. Fortunately this can be easily done in the Nyström framework.

If we compare the Equations (5.3) and (5.8) we see that the correction is performed on a different decomposition of $\hat{\mathbf{S}}$, i.e.:

$$\mathbf{S}_{N,m}\mathbf{S}_{m,m}\mathbf{S}_{N,m}^{\top} = \hat{\mathbf{S}} = \mathbf{C}\mathbf{A}\mathbf{C}^{\top}. \tag{5.9}$$

If we correct $\mathbf{A}$ it is not clear what happens on the left side of the above equation. Therefore, to compute the out-of-sample extension we need to find a simple transformation from one decomposition to the other. Taking a linear part $\hat{\mathbf{S}}_{N,m}$ from the equation 5.9 we get

$$\mathbf{S}_{N,m} = \mathbf{C}_{N,m}\mathbf{A}\mathbf{C}_{m,m}^{\top},$$

which leads after a simple transformation to

$$\mathbf{C}_{N,m} = \mathbf{S}_{N,m}\left(\mathbf{A}\mathbf{C}_{m,m}^{\top}\right)^{-1}.$$

Plugging the above formula into Equation (5.8) we get

$$\begin{aligned}
\hat{\mathbf{S}}^{*} &= \mathbf{S}_{N,m}\left(\mathbf{A}\mathbf{C}_{m,m}^{\top}\right)^{-1}\mathbf{A}^{*}\left(\left(\mathbf{A}\mathbf{C}_{m,m}^{\top}\right)^{-1}\right)^{\top}\mathbf{S}_{N,m}^{\top} \\
&= \mathbf{S}_{N,m}(\mathbf{C}_{m,m}^{\top})^{-1}\mathbf{A}^{-1}\mathbf{A}^{*}\mathbf{A}^{-1}\mathbf{C}_{m,m}^{-1}\mathbf{S}_{N,m}^{\top} \\
&= \mathbf{S}_{N,m}(\mathbf{C}_{m,m}^{\top})^{-1}(\mathbf{A}^{*})^{-1}\mathbf{C}_{m,m}^{-1}\mathbf{S}_{N,m}^{\top} \\
&= \mathbf{S}_{N,m}\left(\mathbf{C}_{m,m}\mathbf{A}^{*}\mathbf{C}_{m,m}^{\top}\right)^{-1}\mathbf{S}_{N,m}^{\top}
\end{aligned}$$

and we see that we simply need to extend the matrix $\mathbf{S}_{N,m}$ by uncorrected similarities between the new points and the landmarks to obtain the full approximated and *corrected* similarity matrix, which then can be used by the algorithms to compute the out-of-sample extension. The same approach can be applied to the dissimilarity matrices. Here we first need to transform the new dissimilarities to similarities using Equation (5.6), correct them and then transform back to dissimilarities.

In [25] a similar approach is taken. First, the whole similarity matrix is corrected by means of a projection matrix. Then this projection matrix is applied to the new data, so that the corrected similarity between old and new data can be computed. This technique is in fact the Nyström approximation, where the whole similarity matrix $\mathbf{S}$ is treated as the approximation matrix $\mathbf{S}_{m,m}$ and the old data, together with the new data build the matrix $\mathbf{S}_{N,m}$. Rewriting this in the Nyström framework makes it clear and more obvious, without the need to compute the projection matrix and with an additional possibility to compute the similarities between the new points.

5.4.4 Proof of concept

We close this section by a small experiment on the ball dataset as proposed in [37]. It is an artificial dataset based on the surface distances of randomly positioned balls of two

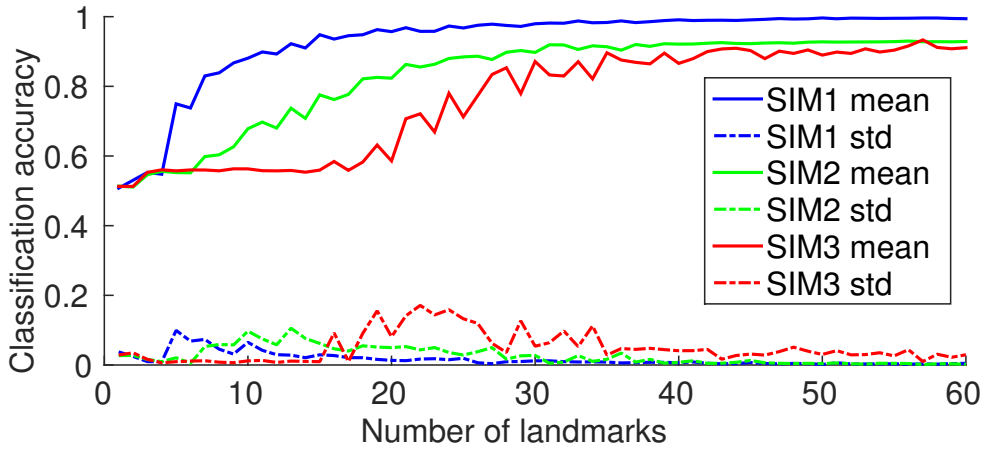


Figure 5.3: Test results for 10 times repeated 10-fold SVM classification on the ball dataset for different number of landmarks using the different encodings.

classes having a slightly different radius. The dataset is non-Euclidean with substantial information encoded in the negative part of the eigenspectrum. We generated the data with 300 samples per class leading to an $N \times N$ dissimilarity matrix $\mathbf{D}$, with $N = 600$.

The data have been processed in five different ways to obtain a valid kernel matrix $\mathbf{S}$. First, the proposed technique was applied, i.e. the matrix $\mathbf{D}$ was approximated by the Nyström technique, then the exact eigenvalue decomposition of the approximated matrix was computed, the eigenvalues were corrected via flipping, resulting in an encoding denoted as $SIM1$, and via clipping, resulting in an encoding $SIM2$. Another encoding was constructed using the landmarks MDS, denoted as $SIM3$. For comparison, two further encodings were constructed by converting $\mathbf{D}$ to $\mathbf{S}$ with double centring, computing the full eigenvalue decomposition and correcting the eigenvalues via flipping, resulting in an encoding denoted as $SIM4$, this approach is also refered to as standard approach in the following, and via clipping, resulting in an encoding $SIM5$. The encodings $SIM1$, $SIM2$ and $SIM3$ can be constructed in linear time, while the encodings $SIM4$ and $SIM5$ have cubic computational complexity. The encodings were processed by a Support Vector Machine in a 10 times repeated 10-fold crossvalidation. The corresponding similarity matrices were used as the kernel matrix, the constraint C, which allows the soft margin, was chosen as 0.2, and least squares formulation was used for optimization.

For the encoding $SIM4$ the SVM was consistently able to achieve perfect classification. For the encoding $SIM5$ the accuracy was 93.00% with the standard deviation of 2.05%. This result shows what already was mentioned earlier: the data contain substantial information in the negative fraction of the eigenspectrum. Accordingly, one may expect that negative eigenvalues should not be removed. This is also reflected in the results for approximated encodings as shown on the Figure 5.3. While flipping-based encoding $SIM1$ is able to achieve almost perfect classification with a sufficient number of landmarks m, the encodings based on clipping tend to an accuracy of only 93% even for large m. Interestingly, there is also a clear difference between both clipping-based techniques. The

accuracy of $SIM2$ increases almost linearly with m, and except for small values of m has low standard deviation. The performance for $SIM3$ on the contrary, stays at almost random classification until a significant amount of landmarks is selected and even then, the variance of the achieved classification is rather high. Thus, the proposed Nyström technique with the exact eigenvalue correction, clearly outperforms the landmark MDS on this data set.

As a last point it should be mentioned that corrections like clipping, flipping and their effect on the data representation are still under discussion and considered to be not always optimal [119]. Additionally, the selection of landmark points is discussed in [177]. Further, for very large data sets (e.g. some 100 million points) the Nyström approximation may still be too costly and some other strategies have to be found, as suggested in [102].

5.5 Experiments

We now apply the priorly derived approach to five non-metric dissimilarity and similarity data and show the effectiveness for a classification task. The considered data are:

- **The SwissProt similarity data** as described in [88] (**DS1**, 10988 samples, 30 classes, imbalanced, signature: $[8488, 2500, 0]$).
- **The chromosome dissimilarity data** taken from [115] (**DS2**, 4200 samples, 21 classes, balanced, signature: $[2258, 1899, 43]$).
- **The proteom dissimilarity data** [36] (**DS3**, 2604 samples, 53 classes, imbalanced, signature: $[1502, 682, 420]$).
- **The Zongker digit dissimilarity data** (**DS4**, 2000 samples, 10 classes, balanced, signature: $[961, 1038, 1]$) from [36] is based on deformable template matching. The dissimilarity measure was computed between 2000 handwritten NIST digits in 10 classes, with 200 entries each, as a result of an iterative optimization of the nonlinear deformation of the grid [78].
- **The Delft gestures dissimilarity data** (**DS5**, 1500 samples, 20 classes, balanced, signature: $[963, 536, 1]$) taken from [36]. This data set is generated from a sign-language interpretation problem. It consists of 1500 samples with 20 classes and 75 samples per class. The gestures are measured by two video cameras observing the positions of the two hands in 75 repetitions of creating 20 different signs. The dissimilarities are computed using a dynamic time warping procedure on the sequence of positions [104].

All datasets are non-metric, multiclass and contain multiple thousand objects, such that a regular eigenvalue correction with a prior double centring for dissimilarity data, as discussed before, is already very costly but can still be calculated to get comparative results.

Table 5.1: Average test set accuracy for SwissProt gestures using a Nyström approximation of 1% and 10% or 30% and no or flip eigenvalue correction.

Landmarks		No	Flip	p-Value
1%	Signature	[109, 1, 10878]	[110, 0, 10878]	
	Accuracy	**92.51** (0.96)	**92.76** (0.66)	0.5966
10%	Signature	[1086, 12, 9890]	[1098, 0, 9890]	
	Accuracy	62.65 (13.60)	**96.85** (0.19)	0.0002
30%	Signature	[3001, 294, 7693]	[3295, 0, 7693]	
	Accuracy	55.73 (9.07)	**96.87** (0.15)	0.0002

Table 5.2: Average test set accuracy for chromosome using a Nyström approximation of 1% and 10% or 30% and no or flip eigenvalue correction.

Landmarks		No	Flip	p-Value
1%	Signature	[41, 1, 4158]	[42, 0, 4158]	
	Accuracy	**91.88** (4.55)	**94.38** (0.35)	0.0814
10%	Signature	[296, 123, 3781]	[419, 0, 3781]	
	Accuracy	38.78 (18.60)	**95.74** (1.77)	0.0002
30%	Signature	[760, 496, 2944]	[1255, 0, 2945]	
	Accuracy	57.34 (10.89)	**96.53** (0.77)	0.0002

5.5.1 Classification performance and matrix approximation accuracy

The data are analysed in two ways, employing either the flipping strategy as an eigenvalue correction, or by not-correcting the eigenvalues[2]. To be effective for the large number of objects we also apply the Nyström approximation as discussed before using a sample rate of $1\%, 10\%, 30\%$[3], by selecting random landmarks from the data. Other sampling strategies have been discussed in [38, 177], also the impact of the Nyström approximation with respect to kernel methods has been discussed recently in [32], but this is out of the focus of the presented approach.

In the first experiment we investigate the influence of N for a fixed number of landmarks on the classification accuracy. In the first case, the kernel matrix is approximated and then corrected using the same 500 landmarks. In the second case, the standard approach is applied. In both cases the size of the data set is increased continuously. The classification results in a 10-fold cross-validation with 10 repeats using the Core-Vector-Machine (CVM) [154] are reported in Figure 5.4. It can be observed, that the proposed approach does not sacrifice classification performance for computational speed.

In the next experiment, different Nyström approximation rates $\approx 1\%, \approx 10\%$ and $\approx 30\%$ are investigated on the uncorrected kernels and on kernels corrected using flip

[2]Clipping and flipping were found similar effective, with a little advantage for flipping. With flipping the information of the negative-eigenvalues is at least somewhat kept in the data representation so we focus on this representation. Shift correction was found to have a negative impact on the model as already discussed in [25].

[3]A larger sample size did not lead to further substantial improvements.

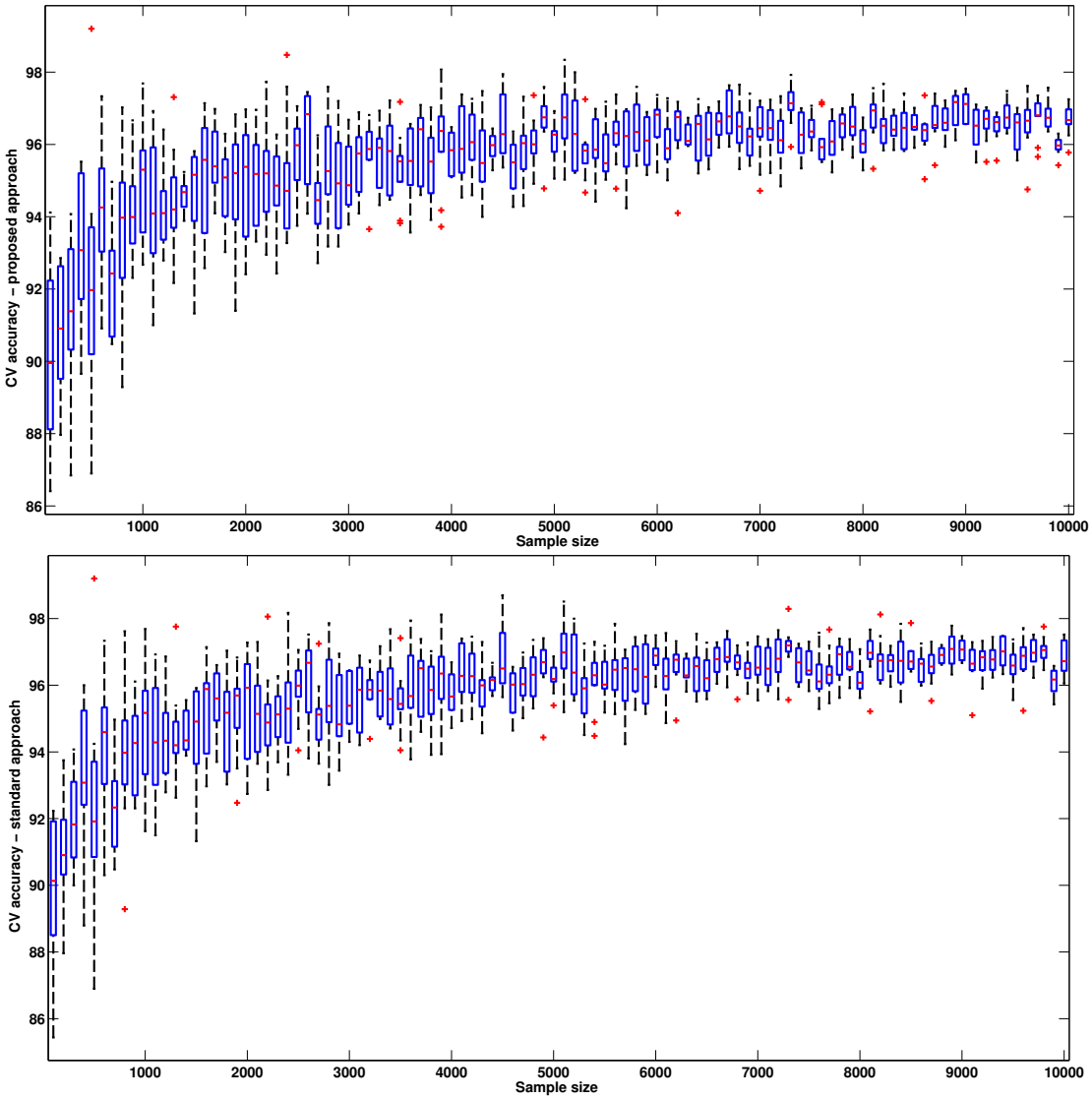


Figure 5.4: Top: box-plots of the classification performance for different sample sizes of DS1 using the proposed approach with 500 landmarks. Bottom: The same experiment but with the standard approach.

Table 5.3: Average test set accuracy for proteom using a Nyström approximation of 1% and 10% or 30% and no or flip eigenvalue correction.

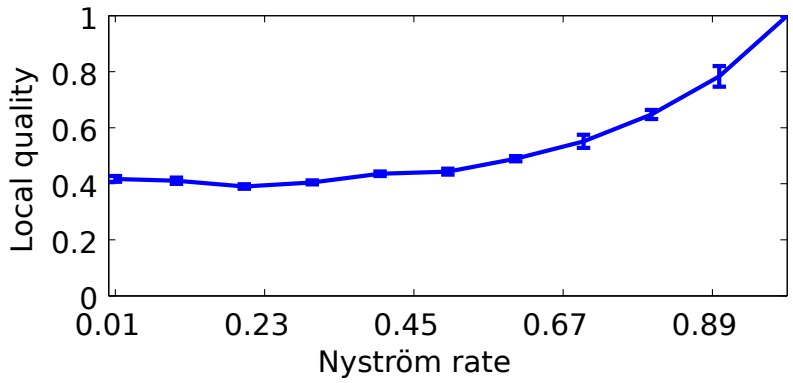
Landmarks		No	Flip	p-Value
1%	Signature	[26, 1, 2577]	[27, 0, 2577]	
	Accuracy	45.69 (13.49)	**84.94** (2.72)	0.0002
10%	Signature	[242, 11, 2351]	[252, 0, 2352]	
	Accuracy	63.18 (24.34)	**96.40** (2.30)	0.0004
30%	Signature	[595, 124, 1885]	[722, 0, 1882]	
	Accuracy	69.44 (14.00)	**97.91** (0.92)	0.0002

Table 5.4: Average test set accuracy for Zongker using a Nyström approximation of 1% and 10% or 30% and no or flip eigenvalue correction.

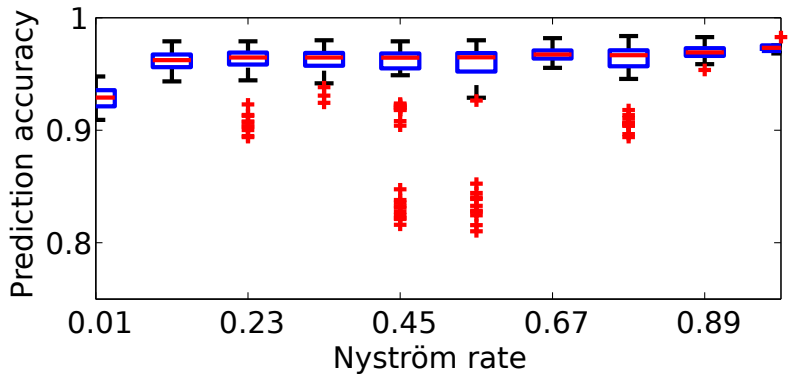
Landmarks		No	Flip	p-Value
1%	Signature	[15, 5, 1980]	[20, 0, 1980]	
	Accuracy	18.95 (11.37)	**84.22** (2.28)	0.0002
10%	Signature	[116, 83, 1801]	[200, 0, 1800]	
	Accuracy	22.30 (5.23)	**93.94** (1.25)	0.0002
30%	Signature	[326, 274, 1400]	[600, 0, 1400]	
	Accuracy	36.85 (3.33)	**93.86** (0.82)	0.0002

Table 5.5: Average test set accuracy for Delft gestures using a Nyström approximation of 1% and 10% or 30% and no or flip eigenvalue correction.

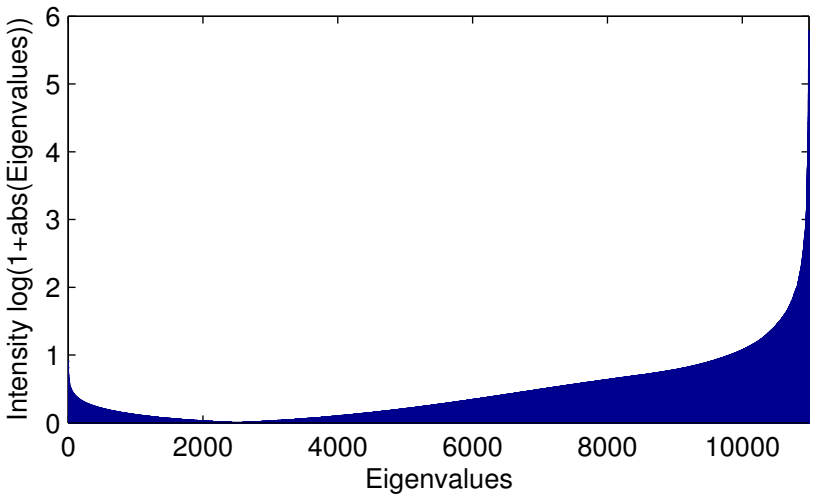
Landmarks		No	Flip	p-Value
1%	Signature	[14, 1, 1485]	[15, 0, 1485]	
	Accuracy	73.07 (11.94)	**87.47** (2.99)	0.0008
10%	Signature	[131, 19, 1350]	[150, 0, 1350]	
	Accuracy	44.55 (22.37)	**95.84** (1.43)	0.0002
30%	Signature	[334, 115, 1051]	[450, 0, 1050]	
	Accuracy	49.35 (7.26)	**92.17** (8.05)	0.0002



(a) SwissProt correlation

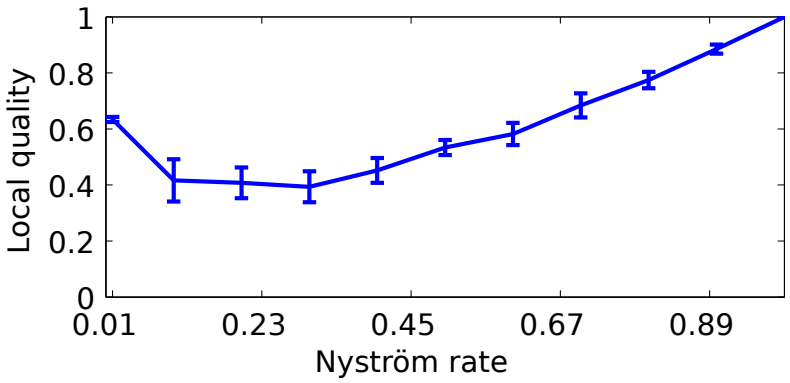


(b) SwissProt accuracy

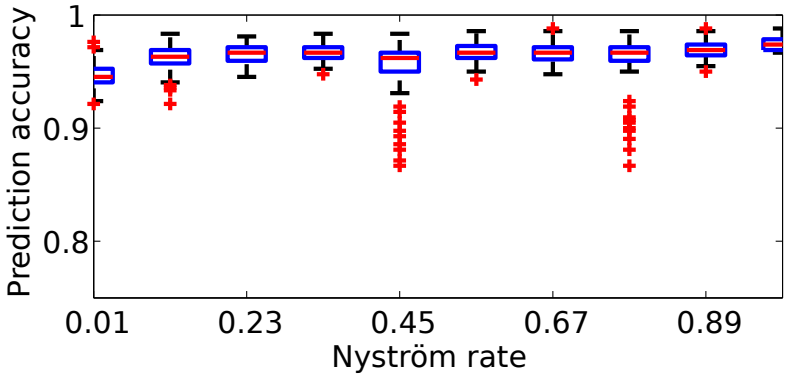


(c) SwissProt eigenspectrum

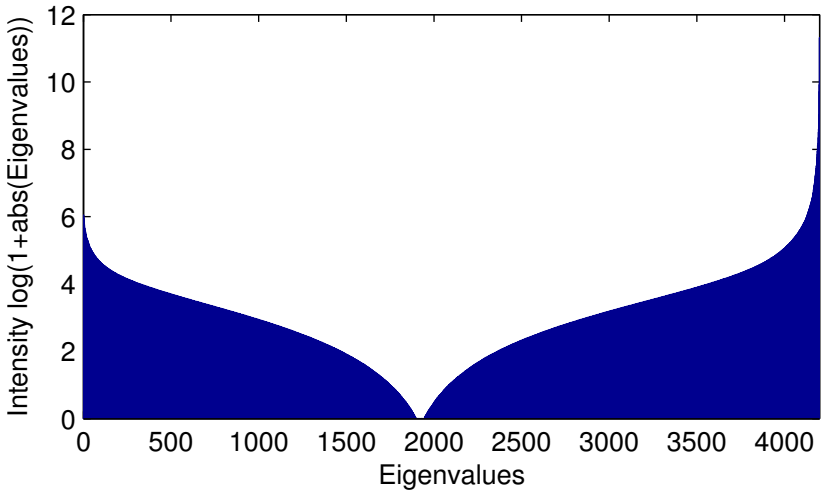
Figure 5.5: Local quality (top) and the cross-validation accuracy (middle) for the SwissProt data set using the proposed approach with an interleaved double centring and Nyström approximation on the dissimilarity data. Bottom: a logarithmic representation of the eigenspectrum of the unapproximated and double centred matrix.



(a) Chromosome correlation

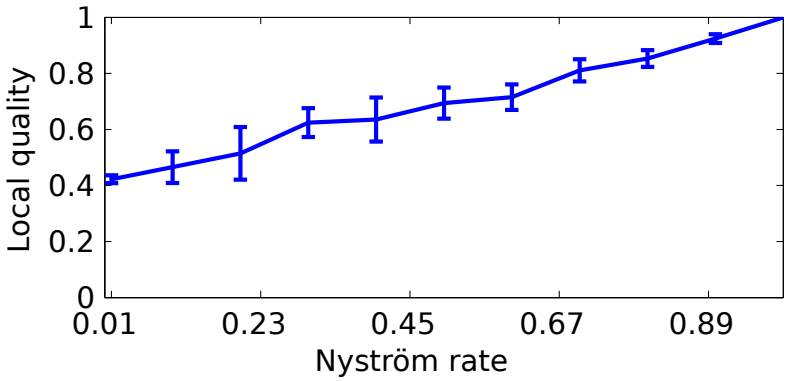


(b) Chromosome accuracy

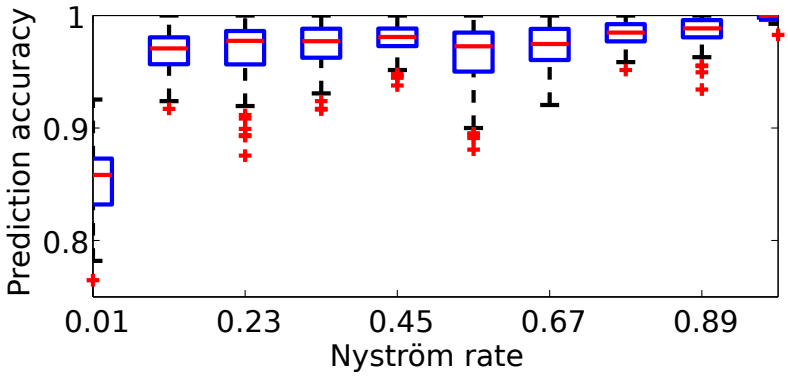


(c) Chromosome eigenspectrum

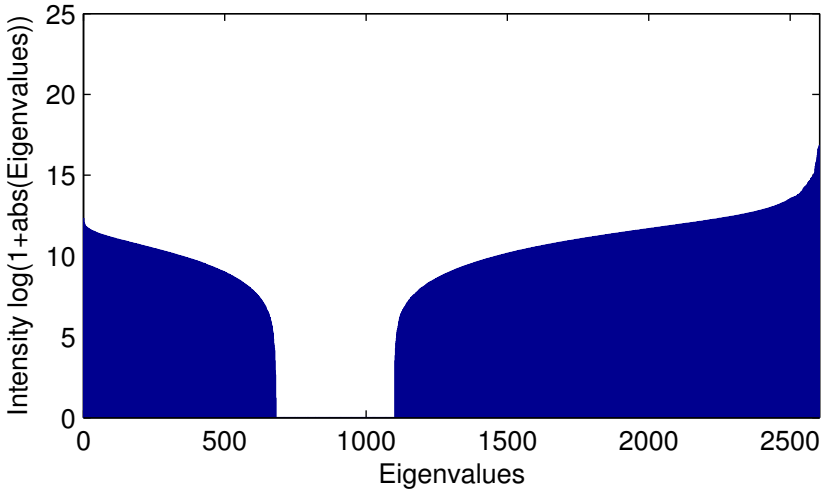
Figure 5.6: Local quality (top) and the cross-validation accuracy (middle) for the chromosome data set using the proposed approach with an interleaved double centring and Nyström approximation on the dissimilarity data. Bottom: a logarithmic representation of the eigenspectrum of the unapproximated and double centred matrix.



(a) Proteom correlation



(b) Proteom accuracy



(c) Proteom eigenspectrum

Figure 5.7: Local quality (top) and the cross-validation accuracy (middle) for the proteom data set using the proposed approach with an interleaved double centring and Nyström approximation on the dissimilarity data. Bottom: a logarithmic representation of the eigenspectrum of the unapproximated and double centred matrix.

Table 5.6: Average test set accuracy for SwissProt (DS1), Chromosome (DS2), Proteom (DS3), Zongker (DS4), Delft gestures (DS5) using the dissimilarity space representation and a linear kernel or an elm kernel.

Data set	linear	elm
DS1	26.01 ± 5.49	72.09 ± 0.96
DS2	76.76 ± 1.11	89.88 ± 0.96
DS3	68.36 ± 2.48	85.37 ± 2.86
DS4	93.70 ± 2.04	95.05 ± 1.71
DS5	87.73 ± 3.83	91.67 ± 2.58

with the proposed technique. Again, classification rates are calculated in a 10-fold cross-validation with 10 repeats using the Core-Vector-Machine (CVM). To compare the results the Wilcoxon rank-sum test is employed and the corresponding p-values are reported. To cancel out the selection bias of the Nyström approximation, the cross-validation does not include a new draw of the landmarks, instead CVM uses the same precomputed kernel matrices. However, our objective is not maximum classification performance (which is only one possible application) but to demonstrate the effectiveness of our approach for dissimilarity data of larger scale. The classification results are summarized in Tables 5.1 to 5.5. First, one observes that the eigenvalue correction has a strong, positive effect on the classification performance consistent with earlier findings [25]. However in case of a small number of landmarks the effect of the eigenvalue correction is less pronounced compared to the uncorrected experiment as shown in Tables 5.1 and 5.2 for DS1 and DS2, respectively. In these cases the Nyström approximation has also reduced the number of non-negative eigenvalues, as shown by the corresponding signatures, such that an implicit eigenvalue correction is obtained. For DS3, see Table 5.3, the remaining negative eigenvalue has a rather high magnitude and a strong impact accordingly, such that the classification performance is sub-optimal for the uncorrected experiment. For DS4, see Table 5.4, the CVM shows particularly bad results for uncorrected input data, indicating that the negative eigenvalues carry important information, but also that the corresponding kernel has a strong negative definite component, which prevents CVM from converging properly. Usually, raising the number of landmarks, in Tables 5.1 to 5.5, improves the classification performance for the experiments with eigenvalue correction. For the experiments without eigenvalue correction however, the performance degenerates instead. This can be explained by the fact, that more and more negative eigenvalues are still kept with raising number of landmarks as shown in the signatures [4].

In Table 5.6 we also show the cross-validation results by use of the priorly mentioned dissimilarity space representation. For simplicity we use an N dimensional feature space and analyse the obtained vector representation by means of a linear kernel and a de facto parameter free elm kernel [42]. For the majority of the experiments the obtained results are significantly worse with the exception of DS4. Also for DS5 a comparison

[4] Comparing signatures at different Nyström approximations also shows that many eigenvalues are close to zero and are sometimes counted as positive, negative or zero.

with a 1% Nyström approximation gives still acceptable results. It should be noted that the results of the elm-kernel experiments are consistently better compared to the linear kernel, indicating the high nonlinearity of the data. Obviously the dissimilarity space representation is in general no reasonable alternative. Additionally it becomes very costly for out-of-sample extensions if the number of considered features becomes large.

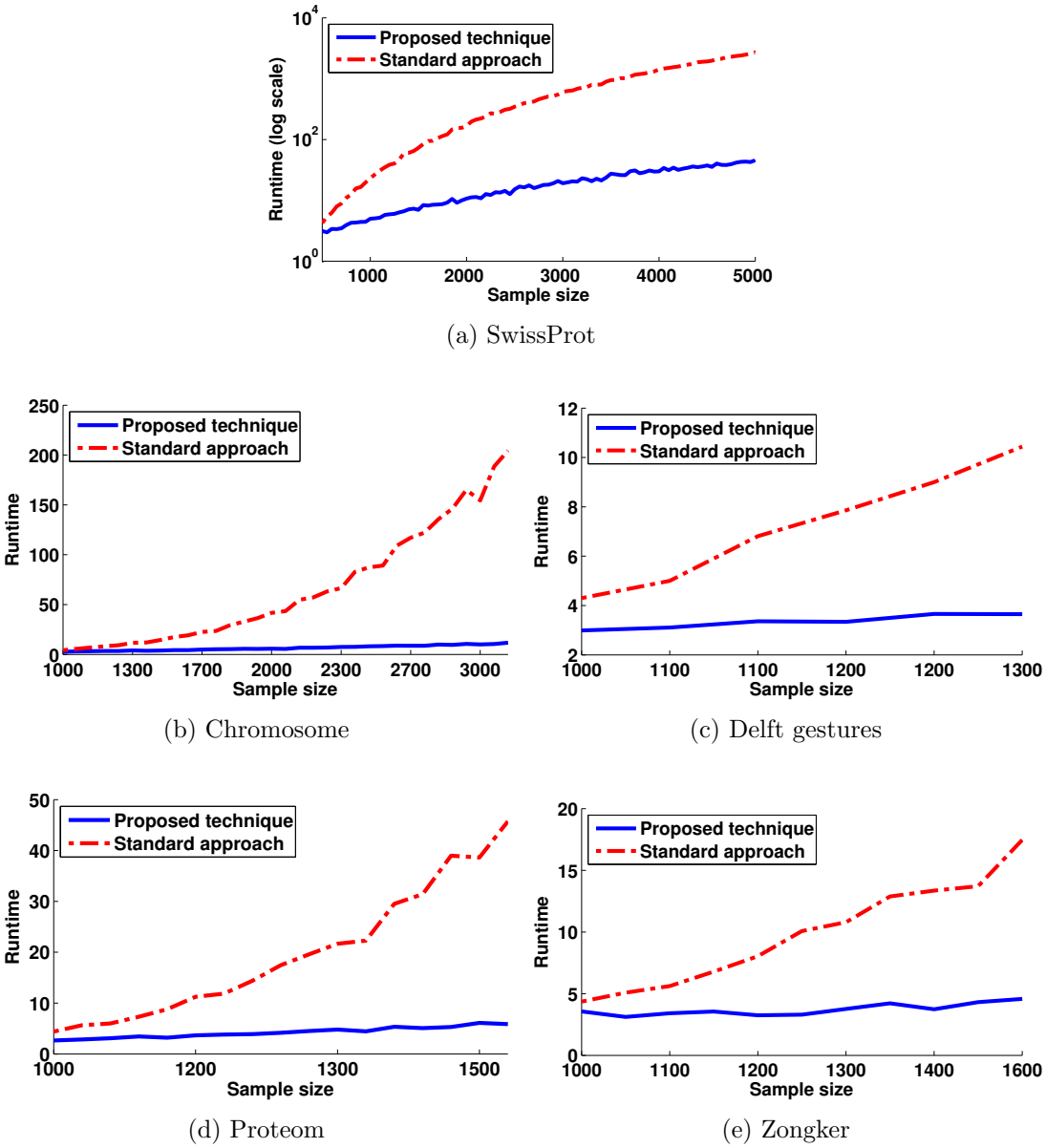
In an another experiment, see Figures 5.5 to 5.7, we analysed the proximity preservation of the approximated and corrected matrix with respect to the unapproximated and corrected matrix. One would expect that for very low Nyström rates (high approximation), only the dominating eigenvalues are kept and the approximation suffers mainly when the eigenspectra are very smooth. At increasing Nyström rates (lower approximation), first more and more small eigenvalues (also negative ones) are kept leading to a more complex data set and accordingly also a more complex proximity preservation task. Finally if the Nyström rates are high (almost no approximation) one would expect a perfect preservation. This effect is indeed observed in Figures 5.5 to 5.7.

We used the evaluation measure Q_{local}, which was proposed in [101] and explained in more detail in section 6.5, to measure how far the local neighbourhoods defined by the proximities (e.g. distances), as calculated by the two approaches, deviate from each other. Low quality values indicate that the neighbourhoods consist of different points in both data sets. Comparing the local quality results (top row in Figures 5.5 to 5.7) with the prediction accuracy on the test data (middle row in Figures 5.5 to 5.7) we see that the neighbourhood preservation in most cases has no effect on accuracy. This agrees with our expectation that the data are potentially clustered and local errors in the data relation have only a weak or no effect on the classification model.

From the analysis we can conclude that, even if the local neighbourhood relations are kept only for approximation rates of above 60%, the classification accuracy is still high for 10% of landmarks. As one can see from smooth eigenspectra in Figures 5.5 to 5.7, the rank of the data sets is rather high, accordingly only for large m the approximation can keep detail information, effecting the local relationships of the data points. Thus, if the different classes are close to each other and have complex nonlinear boundaries, a too small number of landmarks leads to an increased classification error. In practice, as can be seen on the Figures 5.5 to 5.7, the number of the landmarks needs to be very small to have effect on classification. It is thus possible to approximate the matrices by selecting m sufficiently small, without sacrificing the classification accuracy.

5.5.2 Runtime performance

As shown exemplary in Figure 5.4 the classification performance on eigenvalue-corrected data is approximately the same for our proposed strategy and the standard approach. But the runtime performance is drastically better for an increase in the number of samples. To show this we selected subsets from the considered data with different sizes from 1000 to the maximal number, while the number of landmarks is fixed by $L = 500$ and calculated the runtime and classification performance using the CVM classifier in a 10-fold crossvalidation. The eigenvalues have been flipped in this experiment. The results of the proposed approach compared to the standard approach are shown in the plots of Figure 5.8. For larger N the runtime of the standard method (red/dashed line)



(a) SwissProt

(b) Chromosome

(c) Delft gestures

(d) Proteom

(e) Zongker

Figure 5.8: Runtime analysis of the proposed vs the standard approach for the considered dissimilarity data sets. All eigenvalues of the data sets have been processed by flipping.

is two magnitudes larger on log-scale compared to the proposed approach.

5.6 Summary

In this chapter we addressed the analysis of potentially non-metric proximity data and especially the relation between dissimilarity and similarity data. We proposed effective and *accurate* transformations across the different representations. Dedicated learning algorithms for dissimilarities and kernels are now accessible for both types of data. The specific coupling of double centring and Nyström approximation permits to compute an exact eigenvalue decomposition in linear time which is a valuable result for many different methods depending on the exact calculation of eigenvalues and eigenvectors of a proximity matrix. Also the approximation of the data matrix by the Nyström technique improves the complexity of many algorithms to linear one. In particular, this can be applied for relational GTM, as noted in the Table 5.7. It allows not only to process pseudo-Euclidean distances by kernel methods, but also having a corrected euclidean distance matrix is beneficial to relational techniques and can solve different problems, such as convergence issues, as discussed in 4.3.

Table 5.7: Improving the efficiency of relational GTM.

Topic	Relational	Out-of-Sample	Efficiency		Relevance
Technique	Data	Extension	vectorial	relational	Learning
GTM	√	√	√	√	√
t-SNE	√	?	?		?

While our strategy is very effective e.g. to improve supervised learning of non-metric dissimilarities by kernel methods, it is however also limited again by the Nyström approximation, which itself may fail to provide sufficient approximation and accordingly further research in this line is of interest. Nevertheless, dedicated methods for arbitrary proximity data as addressed in [121] will also be subject of future work. For non-psd data the error introduced by the Nyström approximation and the eigenvalue correction is not yet fully understood and bounds similar as proposed in [35] are still an open issue.

This chapter is based on: Andrej Gisbrecht, Alexander Schulz, and Barbara Hammer. Parametric nonlinear dimensionality reduction using kernel t-SNE. *Neurocomputing*, 147:71–82, 2015.

Chapter 6

Parametric nonlinear dimensionality reduction using kernel t-SNE

In the previous chapters we dealt with the generative topographic mapping, as a parametric DR technique. Although we showed, that GTM can be extended to a supervised technique and to relational data in an efficient way, parametric techniques are still limited due to their functional form. E.g. for GTM we assume that the data is distributed along a low dimensional manifold which is embedded in high dimensional space and define our mapping function accordingly. For the real data however, we usually don't know priorly which kind of functions would explain these data in the best way.

Nonparametric DR techniques on the other hand are not limited by a functional form and can produce more powerful and flexible visualizations of high-dimensional data. This property, at the same time, results in a drawback of nonparametric techniques, since they lack an explicit out-of-sample extension. In this chapter, we propose an efficient extension of t-distributed stochastic neighbour embedding (t-SNE), a novel nonparametric DR technique, to a parametric framework, kernel t-SNE, which preserves the flexibility of basic t-SNE, but enables explicit out-of-sample extensions.

In the following, we will first discuss the related work and shortly review popular DR techniques, in particular t-SNE in more detail. Afterwards, we address the question how to enhance nonparametric techniques towards an explicit mapping prescription, emphasizing kernel t-SNE as one particularly flexible approach in this context. Finally, we consider discriminative DR based on the Fisher information, testing this principle in the context of kernel t-SNE.

6.1 Related work

As discussed in chapter 2, most recent DR methods belong to the class of nonparametric 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) even if the techniques pursuit the same goal, such as e.g. neighbourhood preservation, depending on the constructed cost function or how it is optimized, 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 nonparametric 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 [157, 174]. 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 [87]. In addition to this intuitive training procedure and outcome, both techniques have several 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 nonparametric DR techniques, PCA and SOM still by far outnumber these alternatives regarding applications.

In this chapter, to address this gap, we discuss recent developments connected to the question of how to turn nonparametric DR 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 DR. 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 labelling.

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 [30, 98]. These techniques can be extended to nonlinear methods by means of kernelization [6, 106]. 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 [84, 125]: 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 [84, 125, 163]. 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.

6.2 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 ambiguous, since it does not specify which structure of the data should be preserved. Accordingly, 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 [152]: (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 centred 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 computes PCA by learning a mapping function, the latter by optimizing the individual positions of the projected points. 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 DR, which together cover most existing DR approaches: (i) the 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 nonparametric 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, and encoder frameworks such as deep autoencoder networks fall under the first, parametric framework [12, 98, 159]. The second framework can cover diverse modern nonparametric approaches such as Isomap, MVU, LLE, SNE, or t-SNE, as recently demonstrated in the overview [18] and already discussed in chapter 2.

A note on parametric approaches

Parametric approaches are often less flexible as compared to nonparametric 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 modelling of the data. Note that a few alternative manifold learners have been proposed, partially on top of nonparametric 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 [144], manifold charting [15] or corresponding extensions of powerful prototype based techniques such as matrix learning neural gas [1]. Manifold coordination also takes place in parametric extensions of nonparametric approaches such as proposed in locally linear coordination [130]. 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 kernel and relational variants of SOM and GTM [58, 59, 175]. In particular, we already presented relational GTM in chapter 4. Due to their dependence on a full distance matrix, these techniques have inherent quadratic complexity if applied for the full data set. In chapter 5 we discussed the Nyström technique as one possibility to overcome this problem. Unfortunately, in cases where relations between all pairs of points have to be computed explicitly, a low rank representation of a matrix does not result in an improved computational complexity. As an alternative, with an explicit mapping and a corresponding strategy to iteratively train the mapping on parts of the data only, it is possible to reduce 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 [58, 59].

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 [158] 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; σ_i is automatically determined by the method such that the effective number of neighbours coincides with a priorly specified parameter, the perplexity. In the projection space, SNE [74] defines the probabilities in the similar way as in the original space. This leads to the so called crowding problem: in low dimensions it is not enough space to visualise high-dimensional data and unless the data is somehow stretched, several regions of high-dimensional space are mapped on each other. To overcome this problem, t-SNE defines the probabilities in the projection space by the Student t-distribution

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

which has a long tail and allows separation the data. 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 (KL divergence). t-SNE relies on a gradient based optimization technique.

Many alternative nonparametric techniques proposed in the literature have a very similar structure, as pointed out in [18]: 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. [18] summarizes some of today's most popular DR methods this way. In the following, we will exemplarily consider the alternatives maximum variance unfolding (MVU), locally linear embedding (LLE), and Isomap. The rationale 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 neighbourhoods of every point. This problem can be formalized by means of a quadratic optimization problem [169]. LLE represents points in terms of linear combinations of its local neighbourhood 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 [132]. 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 [147].

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 paid 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.

6.3 Kernel t-SNE

How to extend a nonparametric DR 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 nonparametric approaches to a parametric version has been proposed in [18, 46, 156] in different forms. In [156], 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 [18] the principle of plugging a parametric form f_w in any cost function based nonparametric 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 [46], 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 [46]. 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 [46] 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 σ_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, for the points $\mathbf{x}_j \in X'$ the original t-SNE visualisation is retained. 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

Algorithm 2 kernel t-SNE

function KTSNE($\mathbf{X}, nTrain, perpl$)
  $(\mathbf{X}_{tr}, \mathbf{X}_{test})$ = SELECTTRAININGSET($\mathbf{X}, nTrain$)
  $\mathbf{D}_{tr}$ = CALCPAIRWISEDIS($\mathbf{X}_{tr}, \mathbf{X}_{tr}$)
  $\mathbf{D}_{test}$ = CALCPAIRWISEDIS($\mathbf{X}_{tr}, \mathbf{X}_{test}$)
  $\mathbf{Y}_{tr}$ = TSNE($\mathbf{D}_{tr}, perpl$)
  $\boldsymbol{\sigma}$ = 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

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 DR 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 σ_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: σ_i is chosen as a multiple of the distance of $\mathbf{x}_i$ from its closest neighbour in X', where the scaling factor is typically taken as a small positive value. We determine this factor automatically as the smallest value for which all entries of $\mathbf{K}$ can be represented numerically or are inside a predefined interval.

Algorithm 2 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 6.6 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 σ_i parameters for the kernels as described previously.

6.4 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 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?

As it has been proposed in [84, 125, 163] and already demonstrated in chapter 3 nonlinear DR techniques such as the self-organizing map can be enriched 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 [125]. 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 [84].

There are two problems to this approach: first, how to compute this learning metrics efficiently for a given labelled 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 non-parametric estimator

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

Although very simple to implement, this approach has a few important disadvantages. Typically, in a high-dimensional data set, the data is not spread uniformly across the whole space, instead it lies on nonlinear low-dimensional manifolds embedded in this space. The density around a training point will then be concentrated along the directions of the manifold, whereas an isotropic Gaussian centred at the same point would weight all directions equally, thus giving too much probability to the empty space and too little to the manifold [165]. Manifold Parzen Window approach [165] can deal with this problem by learning the covariance matrices associated to each point. Another problem appears in large data sets, since estimation of the probability at a single point requires already $\mathcal{O}(N)$ computations. The usual approach to overcome this problem is to use a sophisticated sampling strategy and select a sub set of the data, which can represent the whole data set trustfully [167]. Here, for the sake of simplicity, the dimensionality of the data is reduced with PCA, and the above formula is used to estimate the probabilities $p(c|\mathbf{x})$ on a small subset of the data.

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 [125]. 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 DR of data can be based thereon.

This approximation allows to compute the Fisher metric, but still the involved costs are quite high. For a data set with N points in D dimensions $\mathcal{O}(N^2)$ distances have to be computed. For each distance there are T intermediate points which should be taken into account. For each such point $\mathbf{x}$ the Fisher information matrix $\mathbf{J}(\mathbf{x})$ and the conditional probability $p(c|\mathbf{x})$ have to be estimated. As stated above, the computation of $p(c|\mathbf{x})$ for a single point is $\mathcal{O}(ND)$ for small amount of classes. The computation of $\mathbf{J}(\mathbf{x})$ requires $\mathcal{O}(D^2)$ for the matrix itself and $\mathcal{O}(ND)$ for the computation of $\mathbf{b}(\mathbf{x}, c)$. Together, this results in $\mathcal{O}(TN^2 \cdot (ND + D^2))$ computational complexity. Alternatively, if the path integrals are required, a graph with all $\mathcal{O}(TN^2)$ intermediate points has to be build. The distance is then computed as the shortest path in this graph, resulting in the computational complexity of $\mathcal{O}(TN^2 \cdot (ND + D^2) + T^3N^6)$. In this case, the Fisher matrices have to be stored, resulting in $\mathcal{O}(TN^2D^2)$ memory complexity. From this calculations it becomes clear, that some sampling strategies as well as preprocessing with PCA become necessary for a large high-dimensional data set.

Efficient integration of the Fisher metric into kernel t-SNE

In [45], 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 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 3 details the resulting procedure. Again, calcPairwiseDis calculates the pairwise Euclidean distance between all points in the given matrices. calcPairwise-FisherDis 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.

6.5 Evaluation measures

DR being ill-posed, it eventually depends on the task at hand which results are considered as optimum. Nevertheless, as motivated in section 2.3, 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

Algorithm 3 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

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 [100]. In this work we will stick to this measure, albeit there are intuitive possibilities to extend this proposal [112].

Here, we do not introduce the full co-ranking matrix as given in [100], rather we restrict to the resulting quantitative value referred to as quality in [100]. Essentially, it is generally accepted that a DR technique should preserve neighbourhoods 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 neighbourhood 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 [100] to determine the k nearest neighbours for every point $\mathbf{x}_i$ in the original space and the k nearest neighbours 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 neighbours 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 neighbourhoods are preserved for a neighbourhood 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 neighbourhood 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 [101] which is based on $Q_m(k)$. First, the neighbourhood size is determined, for which the visualisation has the best quality:

$$k_{max} = \arg\max_k \left(Q_m(k) - \frac{k}{N-1} \right),$$

where from the quality curve of a visualisation the baseline for a random projection was subtracted, and the maximum of the resulting curve was taken as the searched neighbourhood size. All the neighbourhood sizes below k_{max} are then treated as 'local' and the summation over them results in the local quality:

$$Q_{local} = \frac{1}{k_{max}} \sum_{k=1}^{k_{max}} Q_m(k).$$

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 neighbour classification error in the projections.

6.6 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 DR 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 analyse the trade off between size of the training set, required time to compute the projection and the resulting generalization performance of the mapping.
- We analyse 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 [156]. This method has been briefly described in section 6.3.
- 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 $\times$ 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 $\times$ 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 $\times$ 16 gray scale images.

6.6.1 Applying the proposed kernel mapping to various nonparametric dimensionality reduction techniques

The proposed kernel mapping is a general concept for out-of-sample extension and hence applicable to many nonlinear DR 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 DR technique and employ our kernel mapping in order to project the remaining 10,000 data points. In Figure 6.1 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, we use the sub-sampling strategy for quality evaluation, described previously in section 6.5, 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 neighbourhood sizes.

6.6.2 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 and its corresponding out-of-sample extension.

We employ the scalar quality evaluation measure Q_{local} since it allows us to compare the qualities of many projections in a single plot. We also compute the KL divergence

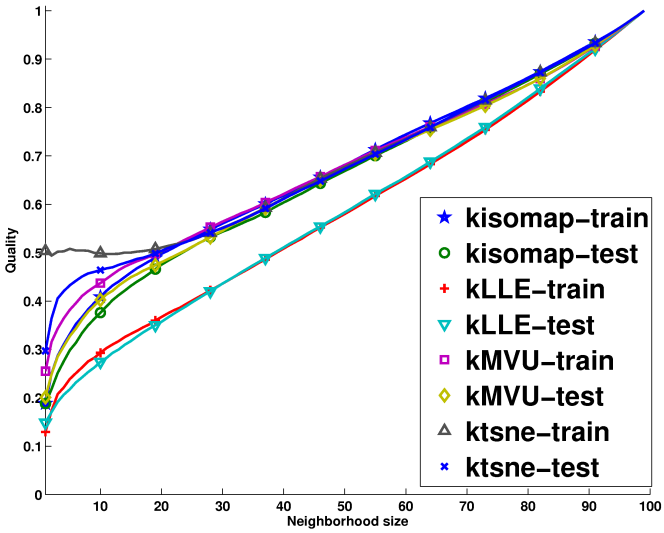


Figure 6.1: Evaluation of various nonlinear dimensionality reduction approaches together with our proposed kernel mapping on the usps data set.

between the points in high- and low-dimensional spaces. Thereby, we calculate 10 projections for each training set and average the resulting values. The results are shown on the Figure 6.2. The local Quality Q_{local} is depicted on the first left axis and referred to as Q_{train} and Q_{test} for training and testing sets, respectively. The KL divergence is depicted on the second left axis and referred to as KL_{train} and KL_{test} for training and testing sets, respectively. In addition, we depict the required running time on the right coordinate axis.

The quality of the projected training set decreases with larger training sets. 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 quadratically. 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}. This can also be supported by an experiment: the quality of the visualisation from ten different subsamples deviated not more than 2%, for training and test sets, respectively. Interestingly, the KL divergence anticorrelates with the Quality Q_{local}. This is because the KL divergence is a part of the cost function evaluated by the quality assessment and it is the reason, why t-SNE along with NeRV achieve so good results according to the quality evaluation.

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

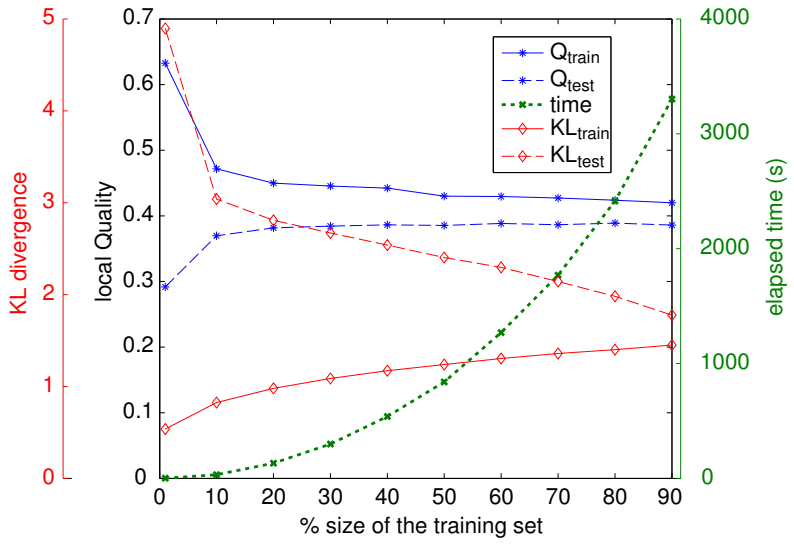


Figure 6.2: Local qualities Q_{local} and required computational time of the projections based on a varying size of the training set.

the distribution of the 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. The values of q_{ij} are grouped into blocks, each block containing probabilities with the value from the specific interval. This way, the horizontal axis shows different blocks and the vertical axis shows the number of probability values in each block. 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 the distribution of the test set below (after flipping horizontally) gives the illustration shown in Figure 6.3. 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 close to zero. From the right we can deduce statements concerning the similarity of both distributions: the amounts of probability values for train set versus test set are very similar for all blocks 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.

6.6.3 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

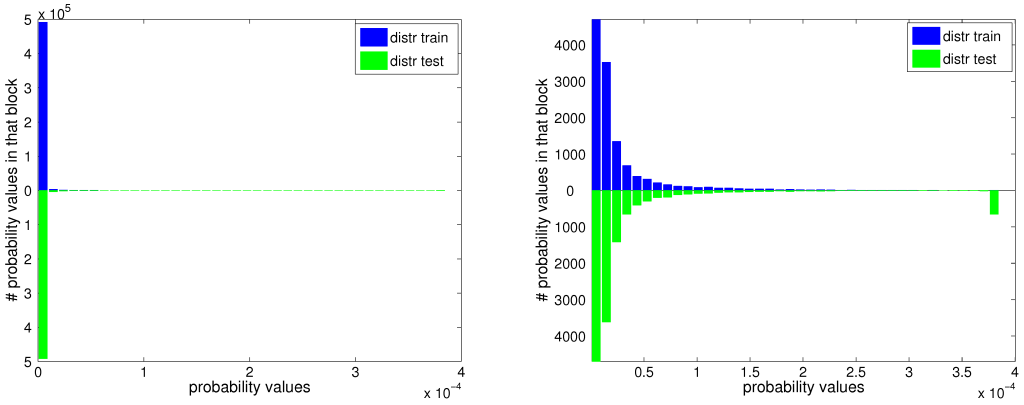


Figure 6.3: Distribution of the probability values q_{ij} as observed in the training set of t-SNE (above zero) and in the out-of-sample extension (below zero after flipping horizontally). The right figure is zoomed in on the y-axis.

for the remaining three data sets 2,000 data 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). The preprocessing was applied to speed up the computation and the number of dimensions was selected in such a way that it does not impair the visualisation quality. For parametric t-SNE the deep architecture of the network, used for this method, realizes a preprocessing step by itself [156]. In preliminary experiments PCA preprocessing had no apparent effect and was therefore left out. 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 6.3.

Figures 6.4 and 6.5 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. The elapsed time includes the preprocessing as well as the training and prediction time, it is shown in Table 6.1. 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 [137].

Further, we apply Fisher kernel t-SNE to obtain visualizations which take the labelling of the data into account. Here we also preprocess the data by projecting them to 30 dimensions. The results are depicted in Figure 6.6.

In order to evaluate the mappings we use the rank based evaluation measure $Q_m(k)$ for different neighbourhood sizes k as described in section 6.5. We use the approximation

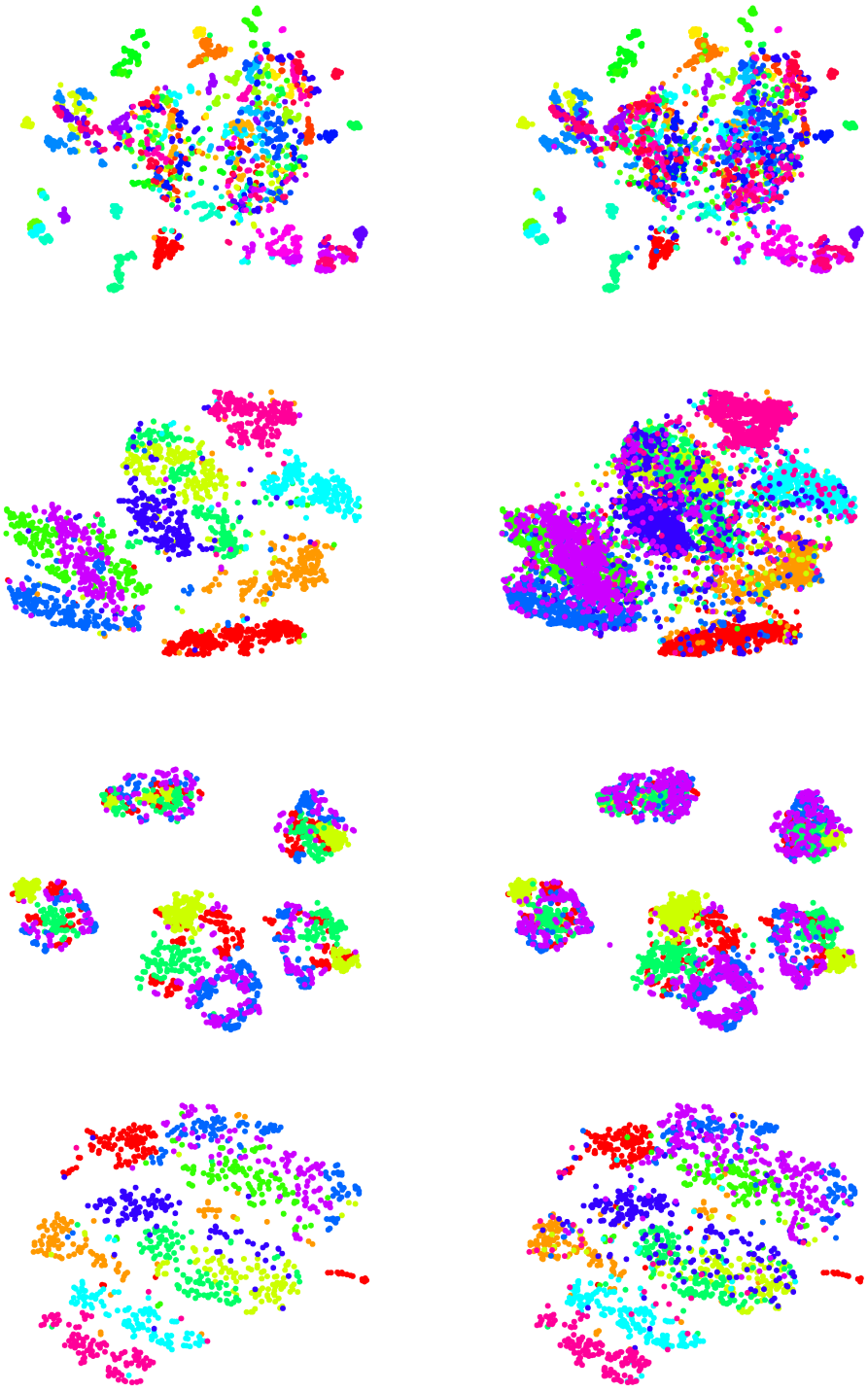


Figure 6.4: Left column: t-SNE applied on the four data sets letter, mnist, norb and usps (from top to bottom). Right column: out-of-sample extension by kernel t-SNE.

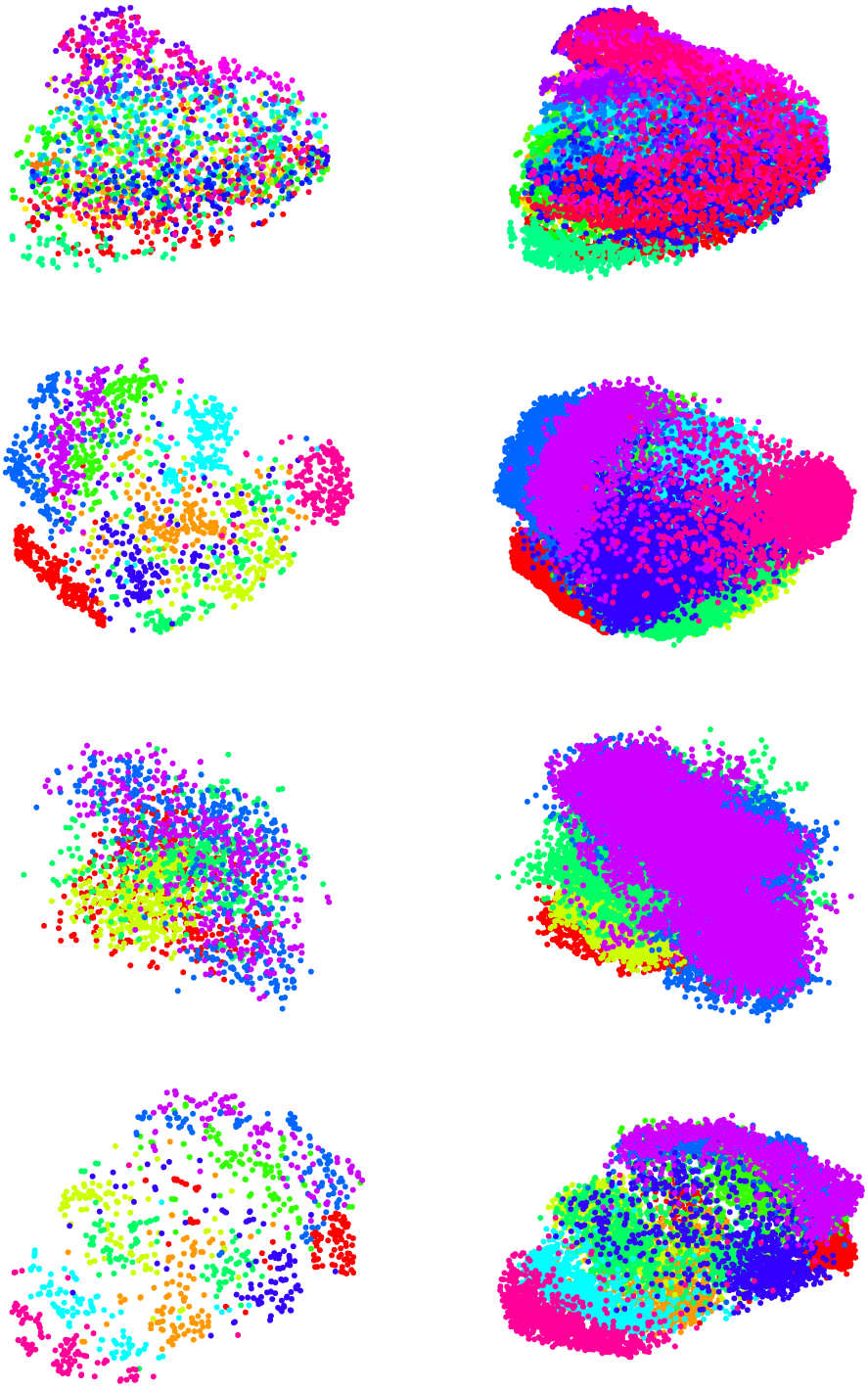


Figure 6.5: Left column: parametric t-SNE mapping learned from the four data sets letter, mnist, norb and usps (from top to bottom). Right column: out-of-sample extension by parametric t-SNE.

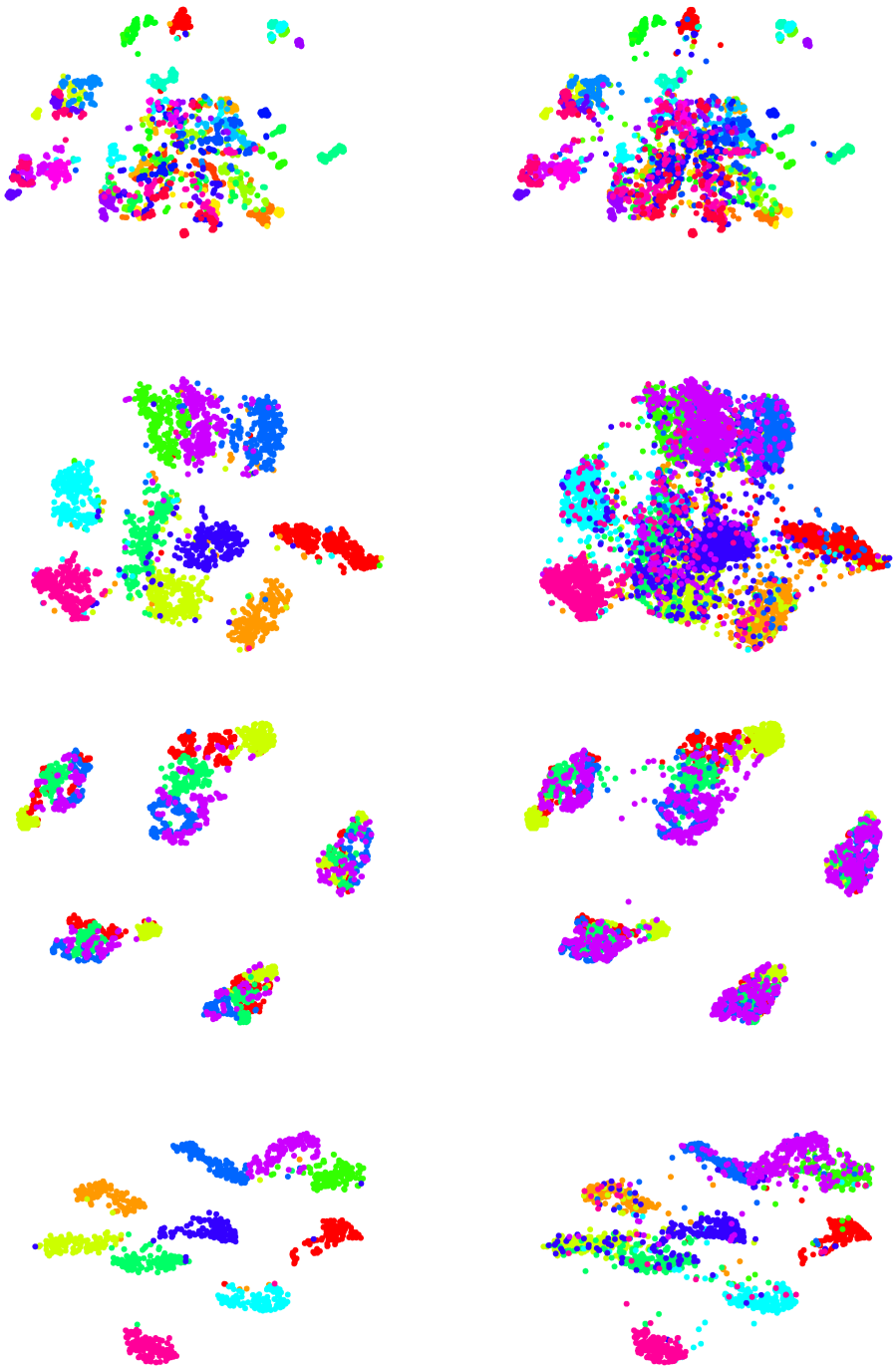


Figure 6.6: Left column: Fisher t-SNE trained on the four data sets letter, mnist, norb and usps (from top to bottom). Right column: out-of-sample extension by Fisher kernel t-SNE.

Table 6.1: 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

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 6.7 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 neighbourhood 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 neighbourhood 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 6.8, 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

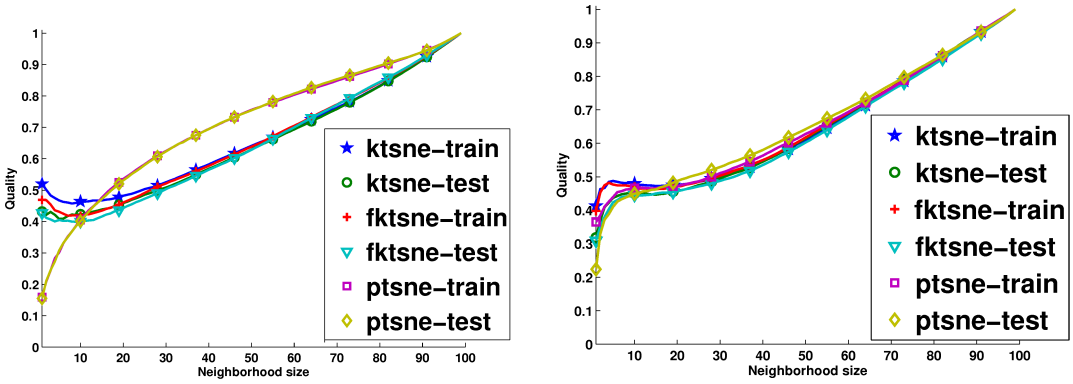


Figure 6.7: Quality curves for the data sets letter (left) and mnist (right).

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 data is necessary to learn the structure of the data properly, 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 6.8 (right). The quality curves of all methods lie close together, while a similar tendency as previously persists: For small neighbourhood 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 neighbourhood ranks. However, this is intended since the methods tries to focus on those changes in the data which affect the labelling of the data. Therefore, a better evaluation for this method would be a supervised evaluation like the k-nearest neighbour classifier described in Section 6.5. Here, we choose $k = 1$. Table 6.2 shows the classification accuracy of the visualizations of all data sets and all methods. Here, 'train' refers to the training set of the DR 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.

6.7 Summary

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

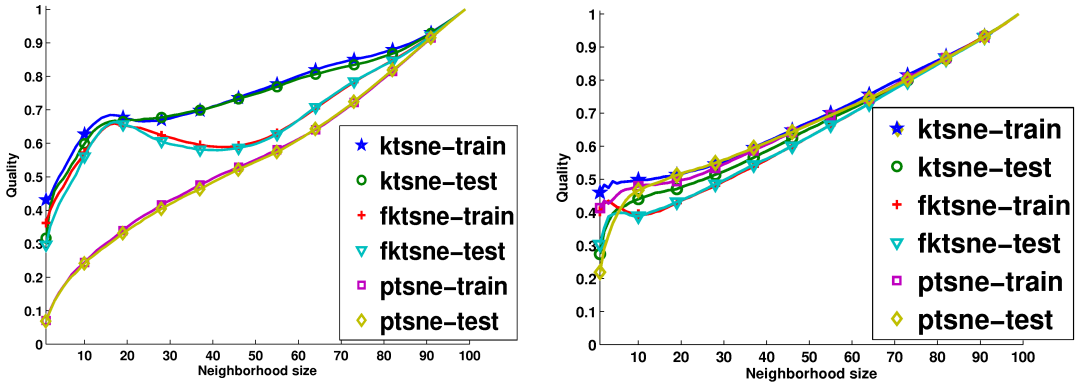


Figure 6.8: Quality curves for the data sets norb (left) and usps (right).

Table 6.2: Accuracies of the nearest neighbour 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%

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. All these improvements are summarized in Table 6.3.

Table 6.3: Benefits achieved by parametric t-SNE.

Topic	Relational	Out-of-Sample	Efficiency		Relevance
Technique	Data	Extension	vectorial	relational	Learning
GTM	√	√	√	√	√
t-SNE	√	**√**	**√**		**√**

The presented approach 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.

Chapter 7

Conclusions

DR is a powerful data mining concept designed to help humans to deal with vast amounts of electronic data. As such, DR should be able to handle large amounts of arbitrarily complex data and produce simple and interpretable results at the same time. In this thesis we presented a variety of techniques which contributed to bridge the gap between existing DR technology and efficient universal methods which can easily be used by practitioners in applications.

In the second chapter we gave an overview on different DR techniques and discussed the categorical separation of the available techniques into parametric and nonparametric ones. We illustrated the typical problems characteristic to each category. Parametric techniques define an explicit mapping from high- to low-dimensional space and are often fast, i.e. they have linear time computational complexity, but since they operate on vectorial data, they can not be applied to complex structured data. Nonparametric techniques, on the other side, directly optimise positions of low-dimensional points and thus are capable of producing a more flexible result. However, this comes with the disadvantage, that they can no longer map new points easily. Further, since all pairs of points are concerned, they frequently have quadratic or even cubic complexity.

We also described relevance learning as a principled way to shape the goal of DR. Generally, it is not possible to produce a low-dimensional projection of high-dimensional data without information loss. Thus, one might formulate the goal of DR as to preserve as much information as possible. Unfortunately, this can be interpreted in an arbitrary way, making DR an inherently ill-posed problem. This formulation gives rise to different approaches, which try to e.g. preserve geodesic distances or neighbourhoods, leading to results which can hardly be shaped by practitioners and their specific demands. By focusing on the aspects of the data which are important according to given auxiliary information we are able to clarify the problem and produce a visualization which is most helpful for the user according to the given auxiliary focus.

In the third and forth chapters we worked with GTM, which is a parametric prototype based technique. We extended it towards supervised visualization by learning the underlying metric. This results in topographic maps which focus on the discriminative information and thus visualise the in-between class relationships. Also, the learned relevance profile allows an insight on importance of different dimensions. In combination with the ability to inspect the prototypes, we obtain a technique with high interpretational

capability.

Although GTM is a powerful technique, there are still shortcomings which limit its applicability. It can not be directly applied to structured data, such as graphs, gene sequences or texts. To overcome this limitation we extended GTM to relational data in chapter 4. Since relational data is characterised by a similarity value for each pair of objects it requires squared amount of space and thus the time complexity of GTM increases from linear to quadratic, making this technique no longer feasible for large data sets. This problem was addressed in chapter 5, where we showed that any symmetric matrix can be approximated by the Nyström technique in linear time. We also gave an in-depth discussion on the nature of the relational data and presented a general approach to convert non-metric (dis-)similarity matrices into metric ones efficiently. This principle allows to process arbitrary data with any proximity based technique, such as e.g. kernel PCA.

In the last chapter we dealt with nonparametric techniques based on the example of t-SNE. It provides a more flexible mapping than a parametric technique, but it is hard to extend an already trained map to new unseen data. Another feature is, that it is already a relational technique, since it operates on similarities between objects. However, due to the same reason, it also has high computational complexity, which makes it problematic for large data sets. To solve this problems we presented a general approach, which allows to extend any DR technique to an explicit mapping and thus map new data in the same way as the training data. It is then possible to train the mapping on a small part of the data for which computational complexity is not yet an issue and extend the mapping in a simple way to the remaining data. To improve the quality of the visualization on a small data set we used an approach for relevance learning as presented in [125]. This technique is too time consuming to apply for a given big data set as a whole, but it is viable on a small data set. Thus, both techniques complement each other and produce good visualization results. These results extend DR methods as proposed in the literature to flexible, user adaptable, generic methods in several respects.

Albeit the developments presented in this thesis solve many important issues, there are still major challenges left which are in the focus of research today. Many of these problems arise when the DR techniques are applied in practice by users who are not experts in machine learning. A variety of user demands or technology in the context of user interaction are still open in DR. We have a short glimpse at a few aspects which arise in this context.

The goal of DR in applications is to visualise data in a way suitable for the user. Often this means, instead of optimising a mathematical cost function only, other objectives should also be concerned which take into account the way in which humans perceive visual information on the one hand, and implicit assumptions of the respective application domain on the other hand. Thus, it is important to analyse, what exactly the users want to see and in which way the visualisation could be most beneficial for them. Relevance learning as introduced in this thesis offers one possibility along this line, further intuitive ways to influence the result are certainly necessary. One interesting approach which has recently been proposed is to incorporate topological indices into the judgement of DR results [107].

The techniques should also be designed in an interactive way, so that the user could

modify the visualisation on the fly, focusing on the aspects important for the given task and gaining more insight into the data by looking at it from different perspectives. To achieve this, the user has to understand how the visualisation is generated and how interaction, i.e. through modification of parameters, affects it. This causes the need for so called white box techniques, which have to be transparent and comprehensible, in contrast to the black box techniques, which are, despite being sophisticated, too complicated for operators who are not familiar with DR. For current DR technologies, for example, the influence of model parameters is not yet fully understood, and an intuitive interactive way to set appropriate meta-parameters is lacking. Related questions are the subject of research in the field of interactive data analysis and visual analytics, for example [62].

Another important issue concerns the reproducibility of visualisation results. Due to randomness in the algorithms, as e.g. in the initialisation, but also if the training set changes slightly, the generated visualisation might differ significantly. This results in an uncertainty for the user regarding which visualisation is the right one. A consistent visualisation would generate more reliable results which could be compared to each other e.g. in slightly different experiments. A related problem is the reliability of the visualisation. Given a visualisation, it is not clear, if it is the only right visualisation for certain data; instead many possibilities exist, each making errors at different places. It would be beneficial to indicate non-trustworthy areas to show, that the underlying structure of the data might be too complicated to visualise. Interesting directions along this line have been proposed in [113], for example.

These challenges, among others, turn visualisation into a a very active research topic with many solutions already discovered, but also with challenges which should be the subject of the future work.

Bibliography

[1] B. Arnonkijpanich, A. Hasenfuss, and B. Hammer. Local matrix learning in clustering and applications for manifold visualization. *Neural Networks*, 23(4):476–486, 2010.

[2] M. Badoiu and K. L. Clarkson. Optimal core-sets for balls. *Comput. Geom.*, 40(1):14–22, 2008.

[3] M.-F. Balcan, A. Blum, and N. Srebro. A theory of learning with similarity functions. *Machine Learning*, 72(1-2):89–112, 2008.

[4] J. E. Barnes and P. Hut. A hierarchical O(n-log-n) force calculation algorithm. *Nature*, 324:446, 1986.

[5] G. D. A. Barreto, A. F. R. Araújo, and S. C. Kremer. A taxonomy for spatiotemporal connectionist networks revisited: The unsupervised case. *Neural Computation*, 15(6):1255–1320, 2003.

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

[7] H.-U. Bauer and K. R. Pawelzik. Quantifying the neighborhood preservation of self-organizing feature maps. *IEEE Transactions on Neural Networks*, 3(4):570–579, 1992.

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

[9] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet. Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering. In *Advances in Neural Information Processing Systems, 16*, 2004.

[10] M. Biehl, B. Hammer, E. Merényi, A. Sperduti, and T. Villmann. *Learning in the context of very high dimensional data (Dagstuhl Seminar 11341)*, volume 1. 2011.

[11] E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2001)*, pages 245–250, 2001.

[12] C. M. Bishop, M. Svensén, and C. K. I. Williams. GTM: the generative topographic mapping. *Neural Computation*, 10(1):215–234, Jan. 1998.

[13] M. Born and V. Fock. Beweis des Adiabatensatzes. *Zeitschrift für Physik*, 51(3-4):165–180, 1928.

[14] R. Boulet, B. Jouve, F. Rossi, and N. Villa. Batch kernel SOM and related Laplacian methods for social network analysis. *Neurocomputing*, 71(7-9):1257–1273, Mar. 2008.

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

[16] G. Brumfiel. High-energy physics: Down the petabyte highway. *Nature*, 469:282–283, 2011.

[17] K. Bunte, M. Biehl, and B. Hammer. Nonlinear discriminative data visualization. In M. Verleysen, editor, *European Symposium on Artificial Neural Networks*, pages 65–70. d-side publications, 2009.

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

[19] K. Bunte, S. Haase, M. Biehl, and T. Villmann. Stochastic neighbor embedding (SNE) for dimension reduction and visualization using arbitrary divergences. *Neurocomputing*, 90:23–45, 2012.

[20] K. Bunte, B. Hammer, and M. Biehl. Nonlinear dimension reduction and visualization of labeled data. In X. Jiang and N. Petkov, editors, *International Conference on Computer Analysis of Images and Patterns*, pages 1162–1170. Springer, 2009.

[21] K. Bunte, B. Hammer, A. Wismueller, and M. Biehl. Adaptive local dissimilarity measures for discriminative dimension reduction of labeled data. *Neurocomputing*, 73(7-9):1074–1092, 2010.

[22] T. J. Buschmann, M. Siegel, J. E. Roy, and E. K. Miller. Neural substrates of cognitive capacity limitations. *PNAS*, 2011.

[23] L. Chen and A. Buja. Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association*, 2009.

[24] W. Chen, Y. Chen, and K. Q. Weinberger. Maximum variance correction with application to A* search. In S. Dasgupta and D. Mcallester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 302–310. JMLR Workshop and Conference Proceedings, 2013.

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

[26] Y. Chen and M. R. Gupta. Fusing similarities and kernels for classification. In *12th International Conference on Information Fusion, FUSION '09, Seattle, Washington, USA, July 6-9, 2009*, pages 474–481, 2009.

[27] Y. Chen, M. R. Gupta, and B. Recht. Learning kernels from indefinite similarities. In *In Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, page 19, 2009.

[28] R. Chitta, R. Jin, T. C. Havens, and A. K. Jain. Approximate kernel k-means: solution to large scale kernel clustering. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 895–903, 2011.

[29] A. Cichocki and S.-I. Amari. Families of alpha- beta- and gamma- divergences: Flexible and robust measures of similarities. *Entropy*, 12(6):1532–1568, 2010.

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

[31] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

[32] C. Cortes, M. Mohri, and A. Talwalkar. On the impact of kernel approximation on learning accuracy. *JMLR - Proceedings Track*, 9:113–120, 2010.

[33] V. de Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Advances in Neural Information Processing Systems*, volume 15, pages 705–712. MIT Press, 2002.

[34] P. Demartines and J. Hérault. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, 8(1):148–154, 1997.

[35] P. Drineas and M. W. Mahoney. On the Nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175, 2005.

[36] R. P. W. Duin. PRTools, march 2012.

[37] R. P. W. Duin and E. Pekalska. Non-Euclidean dissimilarities: Causes and informativeness. In *Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshop, SSPR&SPR 2010, Cesme, Izmir, Turkey, August 18-20, 2010. Proceedings*, pages 324–333, 2010.

[38] A. K. Farahat, A. Ghodsi, and M. S. Kamel. A novel greedy algorithm for Nyström approximation. *JMLR - Proceedings Track*, 15:269–277, 2011.

[39] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[40] J.-C. Fort. Som's mathematics. *Neural Networks*, 19(6-7):812–816, 2006.

[41] M. Francis. Future telescope array drives development of exabyte processing. Ars Technica, 2012.

[42] B. Frénay and M. Verleysen. Parameter-insensitive kernel in extreme learning for non-linear support vector regression. *Neurocomputing*, 74(16):2526–2531, 2011.

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

[44] A. Gisbrecht and B. Hammer. Relevance learning in generative topographic mapping. *Neurocomputing*, 74(9):1359–1371, 2011.

[45] 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*, volume 7619 of *Lecture Notes in Computer Science*, pages 126–138. Springer, 2012.

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

[47] A. Gisbrecht, B. Mokbel, and B. Hammer. The Nystrom approximation for relational generative topographic mappings. In *NIPS workshop on challenges of Data Visualization*, 2010.

[48] A. Gisbrecht, B. Mokbel, and B. Hammer. Linear basis-function t-SNE for fast nonlinear dimensionality reduction. In *IJCNN*, 2013.

[49] A. Gisbrecht, B. Mokbel, F.-M. Schleif, X. Zhu, and B. Hammer. Linear time relational prototype based learning. *Int. J. Neural Syst.*, 22(5), 2012.

[50] A. Gittens and M. W. Mahoney. Revisiting the Nystrom method for improved large-scale machine learning. *CoRR*, abs/1303.1849, 2013.

[51] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems 17*, pages 513–520. MIT Press, 2004.

[52] L. Goldfarb. A unified approach to pattern recognition. *Pattern Recognition*, 17(5):575 – 582, 1984.

[53] A. Gorban, B. Kegl, D. Wunsch, and A. Zinoyev, editors. *Principal Manifolds for Data Visualisation and Dimension Reduction*, volume 58 of *Lecture Notes in Computational Science and Engineering*. Springer, 2007.

[54] A. Gorban and A. Zinovyev. Principal manifolds and graphs in practice: from molecular biology to dynamical systems. *International Journal of Neural Systems*, Vol. 20, No. 3:219–232, 2010.

[55] T. Graepel and K. Obermayer. A stochastic self-organizing map for proximity data. *Neural Computation*, 11(1):139–155, 1999.

[56] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[57] B. Haasdonk and C. Bahlmann. Learning with distance substitution kernels. In C. Rasmussen, H. Bülthoff, B. Schölkopf, and M. Giese, editors, *Pattern Recognition*, volume 3175 of *Lecture Notes in Computer Science*, pages 220–227. Springer Berlin Heidelberg, 2004.

[58] 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.

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

[60] B. Hammer, A. Hasenfuss, F. Rossi, and M. Strickert. Topographic processing of relational data. In *Proceedings of 6th International Workshop on Self-Organizing Maps*, 2007.

[61] B. Hammer and B. J. Jain. Neural methods for non-standard data. In M. Verleysen, editor, *European Symposium on Artificial Neural Networks*, pages 281–292. D-side publications, 2004.

[62] B. Hammer, D. Keim, N. Lawrence, and G. Lebanon. Preface: Intelligent interactive data visualization. *Data Mining and Knowledge Discovery*, 27(1):1–3, 2013.

[63] B. Hammer, A. Micheli, A. Sperduti, and M. Strickert. A general framework for unsupervised processing of structured data. *Neurocomputing*, 57:3–35, 2004.

[64] B. Hammer, A. Micheli, A. Sperduti, and M. Strickert. Recursive self-organizing network models. *Neural Networks*, 17(8-9):1061–1086, 2004.

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

[66] B. Hammer and T. Villmann. Mathematical aspects of neural networks. In M. Verleysen, editor, *Proc. Of European Symposium on Artificial Neural Networks (ESANN'2003)*, pages 59–72, Brussels, Belgium, 2003. d-side.

[67] A. Hasenfuss and B. Hammer. Relational topographic maps. In M. R. Berthold, J. Shawe-Taylor, and N. Lavrac, editors, *Advances in Intelligent Data Analysis VII*, volume 4723 of *Lecture Notes in Computer Science*, pages 93–105, Berlin, 2007. Springer.

[68] T. Hastie and W. Stuetzle. Principal curves. *Journal of Amarican Statistical Association*, 84:502–516, 1989.

[69] R. J. Hathaway and J. C. Bezdek. Nerf c-means: Non-euclidean relational fuzzy clustering. *Pattern Recognition*, 27(3):429–437, 1994.

[70] X. He, D. Cai, S. Yan, and H. Zhang. Neighborhood preserving embedding. In *10th IEEE International Conference on Computer Vision*, pages 1208–1213, 2005.

[71] X. He and P. Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems*, volume 16, 2003.

[72] T. Heskes. Energy functions for self-organizing maps. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 303–315. Elsevier, Amsterdam, 1999.

[73] M. Hilbert and P. López. The world's technological capacity to store, communicate, and compute information. *Science*, 332(6025):60–65, 2011.

[74] G. Hinton and S. Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems 15*, pages 833–840. MIT Press, 2002.

[75] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 2006.

[76] S. Horvath. Aktueller Begriff - Big Data. Wissenschaftliche Dienste des Deutschen Bundestages, November 2013.

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

[78] A. K. Jain and D. Zongker. Representation and recognition of handwritten digits using deformable templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(12):1386–1391, 1997.

[79] M. B. Jones, M. P. Schildhauer, O. Reichman, and S. Bowers. The new bioinformatics: Integrating ecological data from the gene to the biosphere. *Annual Review of Ecology, Evolution, and Systematics*, 37(1):519–544, 2006.

[80] P. Kar and P. Jain. Similarity-based learning via data driven embeddings. In *Proc. of Advances in Neural Information Processing Systems*, pages 1998–2006, 2011.

[81] P. Kar and P. Jain. Supervised learning with similarity functions. In *Proc. of Advances in Neural Information Processing Systems*, pages 215–223, 2012.

[82] S. Kaski. Dimensionality reduction by random mapping: Fast similarity computation for clustering. In *Proceedings of IJCNN'98, International Joint Conference on Neural Networks*, volume 1, pages 413–418. IEEE Service Center, Piscataway, NJ, 1998.

[83] S. Kaski, J. Nikkilä, M. Oja, J. Venna, P. Törönen, and E. Castrén. Trustworthiness and metrics in visualizing similarity of gene expression. *BMC Bioinformatics*, 4(48), 2003.

[84] 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.

[85] A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North. *Information Visualization – Human-Centered Issues and Perspectives*. Volume 4950 of LNCS State-of-the-Art Survey. Springer, 2008.

[86] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

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

[88] T. Kohonen and P. Somervuo. How to make large self-organizing maps for nonvectorial data. *Neural Networks*, 15:945–952, 2002.

[89] E. Kokiopoulou and Y. Saad. Orthogonal neighborhood preserving projections. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 26–30, 2005.

[90] J. Laub. *Non-metric pairwise proximity data*. PhD thesis, 2004.

[91] J. Laub, V. Roth, J. M. Buhmann, and K.-R. Müller. On the information and representation of non-euclidean pairwise data. *Pattern Recognition*, 39(10):1815–1826, 2006.

[92] N. D. Lawrence. Spectral dimensionality reduction via maximum entropy. *Journal of Machine Learning Research - Proceedings Track*, 15:51–59, 2011.

[93] N. D. Lawrence. A unifying probabilistic perspective for spectral dimensionality reduction: insights and new models. *Journal of Machine Learning Research*, 13, 2012.

[94] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[95] J. A. Lee, A. Lendasse, and M. Verleysen. Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis. *Neurocomputing*, 57:49–76, 2004.

[96] J. A. Lee, E. Renard, G. Bernard, P. Dupont, and M. Verleysen. Type 1 and 2 mixtures of Kullback-Leibler divergences as cost functions in dimensionality reduction based on similarity preservation. *Neurocomputing*, 112:92–108, 2013.

[97] J. A. Lee and M. Verleysen. Nonlinear projection with the isotop method. In *ICANN'2002 proceedings - International Conference on Artificial Neural Networks*, pages 933–938, 2002.

[98] J. A. Lee and M. Verleysen. *Nonlinear dimensionality reduction*. Springer, 2007.

[99] J. A. Lee and M. Verleysen. Rank-based quality assessment of nonlinear dimensionality reduction. In *ESANN*, pages 49–54, 2008.

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

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

[102] M. Li, J. T. Kwok, and B.-L. Lu. Making large-scale Nyström approximation possible. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 631–638, 2010.

[103] W.-J. Li, Z. Zhang, and D.-Y. Yeung. Latent wishart processes for relational kernel learning. *JMLR - Proceedings Track*, 5:336–343, 2009.

[104] J. F. Lichtenauer, E. A. Hendriks, and M. J. T. Reinders. Sign language recognition by combining statistical DTW and independent classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):2040–2046, 2008.

[105] S. Liwicki, S. Zafeiriou, G. Tzimiropoulos, and M. Pantic. Efficient online subspace learning with an indefinite kernel for visual tracking and recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 23(10):1624–1636, 2012.

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

[107] M. Maillot, M. Aupetit, and G. Govaert. A generative model that learns betti numbers from a data set. In *ESANN2012, 15th European Symposium on Artificial Neural Networks*, pages 537–542, Bruges, Belgique, 2012.

[108] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. MIT Press, 1982.

[109] R. Memisevic and G. Hinton. Multiple relational embedding. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 913–920. MIT Press, Cambridge, MA, 2005.

[110] H. T. Mevissen and M. Vingron. Quantifying the local reliability of a sequence alignment. *Protein Engineering*, 9(2):127–132, 1996.

[111] B. Mokbel, A. Gisbrecht, and B. Hammer. On the effect of clustering on quality assessment measures for dimensionality reduction. In *NIPS workshop on Challenges of Data Visualization*, 2010.

[112] 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.

[113] B. Mokbel, W. Lueks, A. Gisbrecht, and B. Hammer. Visualizing the quality of dimensionality reduction. *Neurocomputing*, 112:109–123, 2013.

[114] E. Mwebaze, P. Schneider, F.-M. Schleif, J. R. Aduwo, J. A. Quinn, S. Haase, T. Villmann, and M. Biehl. Divergence based classification in learning vector quantization. *Neurocomputing*, 74:1429–1435, 2010.

[115] M. Neuhaus and H. Bunke. Edit distance based kernel functions for structural pattern classification. *Pattern Recognition*, 39(10):1852–1863, 2006.

[116] N. Q. Nguyen, C. K. Abbey, and M. F. Insana. Objective assessment of sonographic: Quality II acquisition information spectrum. *IEEE Transactions on Medical Imaging*, 32(4):691–698, 2013.

[117] E. J. Nyström. Über die praktische Auflösung von Integralgleichungen mit Anwendungen auf Randwertaufgaben. *Acta Mathematica*, 54(1):185–204, 1930.

[118] I. Olier, A. Vellido, and J. Giraldo. Kernel generative topographic mapping. *ESANN*, pages 481–486, 2010.

[119] E. Pekalska and R. P. Duin. *The Dissimilarity Representation for Pattern Recognition. Foundations and Applications*. World Scientific, 2005.

[120] E. Pekalska and R. P. W. Duin. Beyond traditional kernels: Classification in two dissimilarity-based representation spaces. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 38(6):729–744, 2008.

[121] E. Pekalska, R. P. W. Duin, S. Günter, and H. Bunke. On not making dissimilarities Euclidean. In *Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshops, SSPR 2004 and SPR 2004, Lisbon, Portugal, August 18-20, 2004 Proceedings*, pages 1145–1154, 2004.

[122] E. Pekalska and B. Haasdonk. Kernel discriminant analysis for positive definite and indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1017–1032, 2009.

[123] J. Peltonen. *Data exploration with learning metrics*. PhD thesis, Helsinki University of Technology, Dissertations in Computer and Information Science, Report D7, Espoo, Finland, 2004.

[124] J. Peltonen, H. Aidos, and S. Kaski. Supervised nonlinear dimensionality reduction by neighbor retrieval. In *Proceedings of ICASSP 2009, the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1809–1812, 2009.

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

[126] J. Platt. FastMap, MetricMap, and Landmark MDS are all Nyström algorithms, 2005.

[127] E. Renard, P. Dupont, and M. Verleysen. User control for adjusting conflicting objectives in parameter-dependent visualization of data. In *Workshop on Visual Analytics using Multidimensional Projections*, 2013.

[128] H. Ritter, T. Martinetz, and K. Schulten. *Neural Computation and Self-Organizing Maps: An Introduction*. Addison-Wesley, 1992.

[129] V. Roth, J. Laub, M. Kawanabe, and J. M. Buhmann. Optimal cluster preserving embedding of nonmetric proximity data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(12):1540–1551, 2003.

[130] 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.

[131] S. T. Roweis. EM algorithms for PCA and SPCA. In *NIPS*, 1997.

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

[133] F.-M. Schleif, B. Hammer, M. Kostrzewa, and T. Villmann. Exploration of mass-spectrometric data in clinical proteomics using learning vector quantization methods. *Briefings in Bioinformatics*, 9(2):129–143, 2008.

[134] P. Schneider, M. Biehl, and B. Hammer. Adaptive relevance matrices in learning vector quantization. *Neural Computation*, 21:3532–3561, 2009.

[135] P. Schneider, M. Biehl, and B. Hammer. Distance learning in discriminative vector quantization. *Neural Computation*, 21:2942–2969, 2009.

[136] B. Schölkopf, A. J. Smola, and K. R. Müller. Kernel principal component analysis. *Advances in kernel methods: support vector learning*, pages 327–352, 1999.

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

[138] S. Seo and K. Obermayer. Soft learning vector quantization. *Neural Computation*, 15(7):1589–1604, July 2003.

[139] B. Shaw and T. Jebara. Structure preserving embedding. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, 2009.

[140] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis and Discovery*. Cambridge University Press, 2004.

[141] L. Song, A. J. Smola, K. M. Borgwardt, and A. Gretton. Colored maximum variance unfolding. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *NIPS*. MIT Press, 2008.

[142] A. Spitzner and D. Polani. Order parameters for self-organizing maps. In L. Niklasson, M. Boden, and T. Ziemke, editors, *Proceedings of the 8th International Conference on Artificial Neural Networks*, volume 2, pages 517–522. Springer, 1998.

[143] M. Strickert, S. Teichmann, N. Sreenivasulu, and U. Seiffert. High-throughput multi-dimensional scaling (HiT-MDS) for cDNA-array expression data. In W. Duch, J. Kacprzyk, E. Oja, and S. Zadrożny, editors, *Artificial Neural Networks: Biological Inspirations – ICANN 2005*, volume 3696 of *Lecture Notes in Computer Science*, pages 625–633. Springer Berlin Heidelberg, 2005.

[144] 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.

[145] J. A. K. Suykens. Data visualization and dimensionality reduction using kernel maps with a reference point. *IEEE Transactions on Neural Networks*, 19(9):1501–1517, 2008.

[146] J. F. M. Svensén. *GTM: The Generative Topographic Mapping*. PhD thesis, Aston University, 1998.

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

[148] The White House. Big data initiative, 2012.

[149] J. Tian, S. Cui, and P. Reinartz. Building change detection based on satellite stereo imagery and digital surface models. *IEEE Transactions on Geoscience and Remote Sensing*, 2013.

[150] P. Tiňo, A. Kabán, and Y. Sun. A generative probabilistic approach to visualizing sets of symbolic sequences. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–706. ACM, 2004.

[151] P. Tiňo and I. Nabney. Hierarchical GTM: Constructing localized nonlinear projection manifolds in a principled way. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):639–656, May 2002.

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

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

[154] I. W. Tsang, A. Kocsor, and J. T. Kwok. Simpler core vector machines with enclosing balls. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, pages 911–918, 2007.

[155] A. Ultsch and H. P. Siemon. Kohonen's self organizing feature maps for exploratory data analysis. In *Proceedings of the International Neural Network Conference (INNC-90)*, pages 305–308, 1990.

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

[157] L. J. P. van der Maaten. Barnes-Hut-SNE. *CoRR*, abs/1301.3342, 2013.

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

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

[160] T. Vatanen, M. Osmala, T. Raiko, K. Lagus, M. Sysi-Aho, M. Oresic, T. Honkela, and H. Lähdesmäki. Self-organization and missing values in SOM and GTM. *Neurocomputing*, 147:60–70, 2015.

[161] J. Venna. *Dimensionality reduction for Visual Exploration of Similarity Structures*. PhD thesis, Helsinki University of Technology, Espoo, Finland, 2007.

[162] J. Venna and S. Kaski. Local multidimensional scaling. *Neural Networks*, 19:89–99, 2006.

[163] 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.

[164] T. Villmann. *Topologieerhaltung in selbstorganisierenden neuronalen Merkmalskarten*. PhD thesis, University of Leipzig, 1997.

[165] P. Vincent and Y. Bengio. Manifold parzen windows. In *NIPS*, pages 825–832, 2002.

[166] U. von Luxburg, O. Bousquet, and M. Belkin. On the convergence of spectral clustering on random samples: The normalized case. In *Learning Theory, 17th Annual Conference on Learning Theory, COLT 2004, Banff, Canada, July 1-4, 2004, Proceedings*, pages 457–471, 2004.

[167] X. Wang, P. Tiňo, M. A. Fardal, S. Raychaudhury, and A. Babul. Fast parzen window density estimator. In *IJCNN*, pages 3267–3274, 2009.

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

[169] K. Q. 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.

[170] K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision*, 70(1):77–90, 2006.

[171] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research*, 10:207–244, 2009.

[172] D. Werner. *Funktionalanalysis*. Springer-Verlag Berlin Heidelberg, 2011.

[173] C. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.

[174] 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.

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

[176] H. Yin. On multidimensional scaling and the embedding of self-organising maps. *Neural Networks*, 21(2-3):160–169, 2008.

[177] K. Zhang and J. T. Kwok. Clustered Nyström method for large scale manifold learning and dimension reduction. *Neural Networks, IEEE Transactions on*, 21(10):1576–1587, 2010.

[178] K. Zhang, L. Lan, Z. Wang, and F. Moerchen. Scaling up kernel SVM on limited resources: A low-rank linearization approach. *JMLR - Proceedings Track*, 22:1425–1434, 2012.

[179] X. Zhu, A. Gisbrecht, F.-M. Schleif, and B. Hammer. Approximation techniques for clustering dissimilarity data. *Neurocomputing*, 90(0):72 – 84, 2012.