Adaptive dissimilarity measures, dimension reduction and visualization

Bunte, Kerstin

Adaptive Dissimilarity Measures, Dimension Reduction and Visualization

Kerstin Bunte

Book cover: Isosurfaces of different dissimilarity measures on a glass surface:
Front (right to left): Minkowski metric Eq. (2.2) with $p = 1$ and isolevel 0.3 and Itakura-Saito divergence Eq. (9.11) with isolevel 0.2.
Back (right to left): Gamma-divergence Eq. (9.29) with $\gamma = 1.5$ and isolevel 0.02 and Minkowski metric with $p = 3$ and isolevel 0.3.
Alternative and adaptive similarity measures are discussed in this thesis.

Published by *Atto Producties Europe* - www.attoproducties.nl - Groningen

supported by the Netherlands Organisation for Scientific Research (NWO)
under project number 612.066.620

NWO
Netherlands Organisation for Scientific Research

RIJKSUNIVERSITEIT GRONINGEN

Adaptive Dissimilarity Measures, Dimension Reduction and Visualization

Proefschrift

ter verkrijging van het doctoraat in de
Wiskunde en Natuurwetenschappen
aan de Rijksuniversiteit Groningen
op gezag van de
Rector Magnificus, dr. E. Sterken,
in het openbaar te verdedigen op
vrijdag 16 december 2011
om 12.45 uur

door

Kerstin Bunte

geboren op 6 september 1981
te Bielefeld, Duitsland

Promotores:	Prof. dr. M. Biehl
	Prof. dr. N. Petkov
Beoordelingscommissie:	Prof. dr. E. Merényi
	Prof. dr. M. Opper
	Prof. dr. M. Verleysen

ISBN: 978-90-367-5186-5

Contents

Acknowledgments

The last four years leading to this thesis have been an adventurous and pleasant journey. My special thanks goes to my promoter Prof. Dr. Michael Biehl, who gave me the opportunity to come to Groningen and work on this interesting and challenging project. I highly appreciate and admire his qualities as a teacher and excellent supervisor. He always find the right and joyful words to keep me motivated even in difficult times. Furthermore, Michael was always available for discussions and advice. He provides an open environment with many contacts and fruitful grounds for collaboration. Moreover, I like to thank Michael for sharing his little free time for joint activities like the famous mud walking. And last but not least, for letting us experience his excellence as a great connoisseur and chef.

In the same spirit, I thank my second promoter Prof. Dr. Nicolai Petkov for creating a productive atmosphere as the head of the Intelligent Systems group and providing the financial support for conferences and workshops. I am grateful for the valuable support and guidance and for teaching me the keys to become a successful researcher.

I deeply acknowledge my collaborators in Groningen and abroad. Gratefully I thank Prof. Dr. Barbara Hammer for the opportunity to visit her in Bielefeld and benefit from her knowledge and experience. Her enthusiasm is infectious and her nimble persuasion and logical reasoning is highly inspiring and simply: convincing. Similarly, I am thankful to Prof. Dr. M.D. Axel W. E. Wismüller, who invited me for a project at the University of Rochester in the USA. I deeply admire how he combines the workload of a medical doctor and high quality scientific research. Furthermore I am grateful for the collaboration with Prof. Dr. Thomas Villmann, Dr. Frank-Michael Schleif and Sven Haase. They provided me with both: the applicational and the theoretical point of view, vital to Chapters 3, 8 and 9 of the thesis. Many thanks go to my dear colleague Dr. Petra Schneider. Besides our joint publications, crucial for the first part of the thesis, she patiently provided me with all material necessary for a smooth start and answered every question. Moreover I like to thank

my Greek workmate Ioannis Giotis, my doorway to combine the fields of image analysis and machine learning.

During the years of study I met many interesting people. Although we did not work on common projects, I thank Prof. Dr. Michael Wilkinson for his informal help including LaTeX and other scientific issues. With Michael our gatherings were never silent. His wide range of knowledge and strong opinion on every subject fed many lively discussions. My office mates Dr. Petra Schneider, Dr. Aree Witoelar, Ernest Mwebaze and André Offringa provided a nice working environment. I am especially grateful for the sportive, funny and culinary adventurous moments shared with Aree and Petra. Moreover, I would like to thank all my fellow Ph.D. students George Azzopardi, Fred Kiwanuka, Pavel Bulanov, Andrea Pagani, Mahir Can Doganay, Ahmad W. Kamal, Elie Khoury, Victoriya Degeler, Tuan Anh Nguyen and Eirini Kaldeli for our regular lunch sessions and my recent sport mates Ehsan Warriach and Ilche Georgievski. I also appreciated the coffee breaks with Alessandro Crippa, although he always rejected my offers of coffee powder from the USA. My special thanks goes to Ando Emerencia, who challenged me in sport and translated the summary in Dutch for goodwill and a cup of coffee.

Additionally, I'd like to thank the administrative staff from the university for their indispensable support. The secretaries Esmee Elshof, Desiree Hansen, Ineke Schelhaas and Helga Steenhuis were always helpful for sorting through all the paperwork. Alphons Navest, Janieta de Jong, Annette Korringa and Yvonne van der Weerd dealt with administrative issues concerning the Ph.D. study. I thank the CIT Servicedesk and especially Jurjen Bokma and Chris Nossing helping with all system related problems.

Last but not least very special thanks goes to my family, which provided me a loving home and supported my education in all non-scientific ways. My parents always taught me to be interested in the world and to build and fix things ourselves. My father offered me much insight to technical understanding and logical thinking. Although, I am sorry that I lack the patience to really enjoy chess. Moreover, I'd like to thank my mother, the best example of being persistant and assiduous. Albeit she never shared my love for mathematics and computers, I think there is no one in the world more proud of me than her.

Kerstin Bunte
Groningen
October 31, 2011

List of symbols

$\mathcal{X}$	input space	9
n	number of input vectors	9
$\boldsymbol{x}^i$	i-th example	9
y^i	i-th label	9
N	dimensionality of the data	9
C	number of classes	9
$\boldsymbol{w}^j$	j-th prototype	9
$c(\boldsymbol{w}^j)$	class of j-th prototype	9
$n_{\boldsymbol{w}}$	number of prototypes	9
d	dissimilarity measure	9
R^i	receptive field of prototype $\boldsymbol{w}^i$	9
E	cost function	11
Φ	scaling function	11
μ	relative difference distance	11
$\boldsymbol{\lambda}$	relevance vector	13
$\boldsymbol{\lambda}^j$	local relevance vector	14
Λ	relevance matrix (GMLVQ)	14
Ω	self-affine transformation (GMLVQ)	15
Λ^j	local relevance matrix (LGMLVQ)	17
Ω^j	local transformation (LGMLVQ)	17
κ	number of target neighbors (LMNN)	18
Y	matrix indicating same class memberships (LMNN)	19
X	matrix containing the perimeter of invasion (LMNN)	19
Γ	relevance matrix (LMNN)	18
Υ	transformation matrix (LMNN)	58
M	target dimension of dimension reduction	22

List of Figures

List of Algorithms

Chapter 1

Introduction

Due to advanced sensor technology, rapidly increasing digitalization capabilities and the availability of less and less expensive storage volume the amount of data has grown tremendously in the last decades. In the years between 1999 and 2002 an increase of stored information about 30% each year was estimated (Lyman and Varian 2003). Usually this data consists of a variety of measured features leading to also very high dimensional data sets. Manually inspection of the data becomes more costly and automatic methods to help humans to quickly scan through massive data amounts are desirable. This gave rise to many applications in computer science to process the available data: advanced techniques including data mining (Han and Kamber 2005), pattern recognition (Duda et al. 2000) and machine learning (Mitchell 1997, Ripley 1996, Bishop 2006), among others. Even with great progress in those fields the optimization of existing methods and development of novel schemes is highly desirable to perform faster and more efficient data analysis.

The field of machine learning concerns the design of algorithms, which aim at the optimization of adaptive systems on the basis of example data. A model is adapted to learn complex patterns and process new data coming from the same domain better regarding the specified objective. The analysis of patterns involves a number of tasks including data representation, classification, clustering, density estimation, regression, feature extraction and dimension reduction, just to name a few. A lot of data visualization tools have been developed to use cognitive capabilities of humans for structure detection in visual images. Structural characteristics of the data can be captured almost instantly by humans despite the amount of data points which are represented in the visualization. Hence, dimension reduction and visualization are commonly used modern data mining techniques (Lee and Verleysen 2007). Machine learning is broadly categorized into reinforcement, supervised and unsupervised learning. Reinforcement learning is inspired by behaviorist psychology and concerns the finding of suitable actions to maximize some notion of reward (Sutton and Barto 1998). Supervised techniques involve external supervision, which provides correct responses to the given inputs. The aim is usually the discrimination of the categories and to maximize the generalization for novel data. Unsupervised methods, on the other hand, do not need supervision and their goal is the discov-

ery of underlying structures and regularities based on the definition of some basic properties of the data. An elaborate description concerning the history of machine learning can be found in, e. g. (Bishop 1995, Ripley 1996, Mitchell 1997, Duda et al. 2000, Bishop 2006).

A very intuitive supervised technique called k-Nearest Neighbor (k-NN) classifier compares the unknown data to all known examples with respect to some dissimilarity measure (Duda et al. 2000). Obviously the computational effort and memory usage scales with the number of known samples. Therefore prototype-based techniques were developed, which employ representations of data subsets. The prototypes are vector locations in the feature space. They usually serve as typical representatives and reflect the characteristics of the data in their direct neighborhood. Some prominent unsupervised examples are the Self-organizing Map (SOM) (Kohonen et al. 2001) and Neural Gas (NG) (Martinetz and Schulten 1991). And a popular supervised family of such prototype-based classification methods is Learning Vector Quantization (LVQ) (Kohonen et al. 2001). All these methods crucially depend on the distance measure, which is used to adapt the prototype positions and performs the nearest prototype classification. Therefore the learning of adaptive metrics with respect to the given problem at hand was investigated (Xing et al. 2002, Chopra et al. 2005, Frome et al. 2007, Schneider et al. 2009b, Schneider et al. 2009a).

This thesis investigates adaptive dissimilarities and applications varying from classification up to supervised and unsupervised dimension reduction.

1.1 Scope of this thesis

The objective of this thesis is manifold, it contains:

- the introduction of prototype-based adaptive dissimilarity learning with limited rank matrices,
- a new method based on that principle for learning in complex valued data domains and
- a general view and new algorithms for unsupervised as well as supervised dimension reduction and visualization.

Adaptive dissimilarities are a powerful tool, which are shown to improve the performance of supervised methods, such as for example LVQ and the k-NN classifiers. These classification algorithms crucially depend on the distance measure used. Metric adaptation techniques allow the learning of discriminative dissimilarity mea-

sures from a given set of representative example data. Restrictions in adaptive matrix learning, e. g. the limitation of the rank, enables the learning of discriminative global or local linear transformations. These transformations can then be used for supervised dimension reduction and visualization. It also reduces the number of the effective learning parameters, which might be interesting from the computational point of view.

In the first part of this contribution previously proposed methods for metric learning in LVQ are extended to limited rank matrices. Several practical applications are investigated including Content Based Image Retrieval (CBIR), dimension reduction and visualization. Furthermore we provide an extension which can be used on complex valued data shown on an example for texture classification in images.

The second part of this thesis focuses on dimension reduction and visualization. We provide a general view on existing dimension reduction methods, which originally provide just an implicit mapping of the given data points itself. Based on this general principle we extend these methods to learn the parameters of explicit mapping functions instead. This provides direct out-of-sample extensions, reduces computational effort by restricting the learning process just on a small subset of the possible large data set and enables the formal investigation of the generalization ability. Furthermore we provide an unsupervised dimension reduction method, which in contrast to other techniques exhibit a complexity which scales linear with the number of data points in every step. It aims in the combination of fast online learning with the high quality of direct divergence optimization, successfully used by state-of-the-art techniques.

1.2 Outline

This section briefly addresses the outline of the thesis and the topics of the chapters. The thesis is divided into two parts. Part I spans from Chapters 2 to 4 and discusses adaptive dissimilarity measures especially as extensions of LVQ. The metric learning defined in this work can be reformulated to learn global or local linear projections of the data, which smoothly leads over to Part II of the thesis dealing with dimension reduction.

The chapters are organized as follows: Chapter 2 provides a short introduction to prototype-based learning and adaptive dissimilarities. Basic algorithms like Generalized LVQ (GLVQ) and Generalized Matrix LVQ (GMLVQ) are described in detail. The metric adaptation scheme is then modified to use limited rank matrices, which reduce the number of parameters and thus the computational effort and gives di-

rect access to supervised dimension reduction. The latter aspect is resumed and investigated in more detail in Chapter 7 in the second part of this thesis.

In Chapter 4 adaptive dissimilarity learning is used in an application for CBIR in Dermatology. The aim is a computer aided diagnosis system which helps the user, e. g. medical doctors, with targeted searches in image data bases. A learned discriminative distance measure is used to retrieve an arbitrary number of most similar pictures from a data base of images of skin lesions. Two methods for metric learning are used and compared: Large Margin Nearest Neighbor (LMNN), which bases on the k-NN algorithm, and the LVQ based approach. It is shown, that adaptive dissimilarities can be used to improve the performance of a CBIR system.

Chapter 5 introduces a variant of LVQ defined on complex valued data. The modification is shown on one example application for texture classification in color images. These variant called Color Image Analysis LVQ (CIA LVQ) combines well known image analysis filter techniques with prototype-based transformation learning defined in the Fourier domain.

Chapter 6 provides an introduction to the second part of the thesis: dimension reduction and visualization. An overview over existing techniques is given and a general principle is formulated. Based on that principle a general framework is proposed which extends given dimension reduction techniques to learn an explicit mapping function. This way those methods, which are originally introduced to provide implicit point-to-point embeddings can be extended to learn mapping functions instead. Out-of-Sample extensions become immediate, the investigation of the generalization ability is possible and it can save computational effort, because the mapping function can be learned on a representative small subset of the data.

In Chapter 7 the adaptive distances and discriminative transformations introduced in Chapter 2 are used for supervised dimension reduction and visualization. A variety of given unsupervised techniques are extended to use label information by plugging in the supervised learned distance or the local linear transformations.

Most dimension reduction techniques preserve properties extracted from local neighborhoods. This requires the computation of pairwise distances, so the computational effort squares with the number of points. Chapter 8 introduces a dimension reduction method which combines the high performance of direct divergence optimization with fast online learning, leading to a complexity growing linear with the number of points. There are numerous divergences offering different properties. Chapter 9 gives an overview over the three divergence families and examples thereof. Using the concept of Fréchet derivatives three algorithms are expanded to the use of arbitrary divergences.

Finally, Chapter 10 presents a brief summary of the research and a collection of ideas for future work and investigation.

Part I

Adaptive Dissimilarity Measures

Chapter 2

Distance Based Classification

Everything has its beauty but not everyone sees it.

Confucius

Abstract

This chapter introduces the basic Learning Vector Quantization (LVQ) algorithms and notations used throughout the thesis. We discuss nearest prototype classification and a set of LVQ learning schemes, which are relevant in the context of this work. Furthermore we explain the concept of parameterized dissimilarity and metric adaptation proposed in the literature.

2.1 Introduction

Machine learning (Mitchell 1997, Bishop 2006) constitutes a huge field in computer science expanding into broad distribution of both, application and theory. The term "learning" comprises the biological point of view by modeling the theory of psychologists of learning in animals and humans. And it also addresses the development of algorithms aiming at the adjustment to a given objective based on empirical data. Thus, from a given set of input/output pairs produced by an complicated unknown process a machine should be able to adjust its internal structure such that the correct output is reproduced for a large number of samples. This part of the thesis concentrates a subfield usually referred to as supervised learning: Samples are given for which the output is (sometimes only approximately) known. The aim is to find a hypothesis that closely agrees with these given data and generalizes well, i.e. produces the desired output also for new samples.

Learning Vector Quantization (LVQ) and its variants constitute a popular family of supervised prototype-based classifiers. The basic algorithm introduced by (Kohonen 1986) is parameterized by a set of labeled prototypes representing the classes in the input space in combination with a dissimilarity measure. The classification takes places by a nearest prototype scheme, i.e. a new sample is assigned to the class represented by the closest prototype with respect to the given metric.

These algorithms are naturally suitable for multi-class problems without changing the learning rules and the complexity is usually dependent on the number of prototypes and only indirect on the number of classes. This classification procedure is closely related to the popular k-Nearest Neighbor (k-NN) approach (Cover and Hart 1967), which keeps the given labeled data set as a reference set and classifies every new data point to the class given by the majority among its k nearest neighbors. Although the k-NN approach is one of the most intuitive and simplest classification algorithms it shows often very good performance. Nevertheless, it might become very expensive in memory usage and computation for very large reference sets. Prototype methods overcome those problems by defining a clustering on the data. Another advantage of LVQ is the interpretability of the resulting parameters: It does not suffer from a "black box" character like an Artificial Neural Network (ANN) or a Support Vector Machine (SVM). The prototypes reflect the characteristic class-specific attributes of the input samples.

The basic heuristic algorithm, called LVQ1 (Kohonen 1986), adapts a set of prototypes from labeled training data by implementing Hebbian learning steps. Additionally, Kohonen introduced two alternative learning schemes: optimized learning-rate LVQ (OLVQ1) and LVQ2.1, aiming at faster convergence and better approximation of Bayesian decision boundaries, respectively. Furthermore, several LVQ variants were proposed, which are derived from an explicit cost function (Sato and Yamada 1996, Seo and Obermayer 2002, Seo et al. 2003). Cost function based approaches are easily extended to a larger number of adaptive parameters. And methods of theoretical learning theory can be used to investigate risk bounds and convergence behavior. A mathematical analysis with respect to the cost function is performed in (Sato and Yamada 1998) and the authors of (Crammer et al. 2002) showed that LVQ aims at margin optimization and therefore good generalization ability can be expected. Further theoretical analysis of different LVQ variants and statistical physics investigations on simplified model situations can be found in (Ghosh et al. 2006, Biehl et al. 2007). Further extensions of the LVQ classification scheme includes the combination with other prototype-based learning schemes. For example the comprehension of the neighborhood cooperation known from Self-organizing Map (SOM) or Neural Gas (NG) into the learning process (Kohonen 2002, Hammer, Strickert and Villmann 2005b).

Particularly interesting for distance-based machine learning methods like mentioned before is the employed dissimilarity measure. A very common choice is the Euclidean distance, which is a special case of the Minkowski metric. Recently, also divergences known from information theory were used as dissimilarity measure in vector quantization schemes (Mwebaze et al. 2011, Villmann and Haase 2011). In supervised settings where auxiliary information, such as labels, is available the

adaptation of the distance by means of metric learning became popular. Some LVQ variants have been proposed, which aim at the optimization of the distance measure for a specific application (Bojer et al. 2001, Hammer and Villmann 2002, Schneider et al. 2009b, Schneider et al. 2009a). Also methods which aim at the optimization of the k-NN classification scheme have been developed using adaptive dissimilarities (Goldberger et al. 2004, Weinberger et al. 2006). Usually a big improvement of the classification performance can be observed when metric learning is incorporated in the algorithms. In the following section we will review some machine learning techniques used throughout the thesis, especially, existing metric adaptation schemes are presented.

2.2 Nearest prototype classification

We assume that the input data $\mathcal{X}$ consists of n examples $\{\boldsymbol{x}^i\}_{i=1}^n \in \mathbb{R}^N$ together with their corresponding labels $y^i \in \{1, \ldots, C\}$, where N denotes the dimension and C the number of classes or categories. A nearest prototype classifier is parameterized by a set of labeled prototype vectors $\boldsymbol{w}^j$, also called *codebook*, and a distance measure d. The protoytpes $\boldsymbol{w}^j$ are defined on the same feature space as the input data and they carry the label $c(\boldsymbol{w}^j)$ of the class they aim to represent. This implies the definition

$$\mathbf{W} = \{(\boldsymbol{w}^j, c(\boldsymbol{w}^j)) \in \mathbb{R}^N \times \{1, \ldots, C\}\}_{j=1}^{n_{\boldsymbol{w}}} \ , \tag{2.1}$$

where the number of prototypes $n_{\boldsymbol{w}} \geqslant C$, which means that at least one prototype per class is needed. A popular distance measure is the Euclidean distance, which is a special case of the general Minkowski metric

$$d^p(\boldsymbol{x}, \boldsymbol{w}) = \left(\sum_{i=1}^{N} |x_i - w_i|^p \right)^{\frac{1}{p}} \tag{2.2}$$

with $p = 2$. Examples of the equidistance lines using the Minkowski metric and different values for p are shown in Figure 2.1. The classification takes places by a winner-takes-all scheme, i.e. a new data point $\boldsymbol{x}$ is assigned to the class represented by the closest prototype:

$$\boldsymbol{x} \leftarrow c(\boldsymbol{w}^i), \text{ with } \boldsymbol{w}^i = \arg\min_j d(\boldsymbol{x}, \boldsymbol{w}^j), \tag{2.3}$$

braking ties arbitrary. The set of protoytpes and the metric is partitioning the input data space. Each prototype $\boldsymbol{w}^i$ has a receptive field R^i, which is a region in the feature space where $\boldsymbol{w}^i$ is closer to the data than any other prototype:

$$R^i = \{\boldsymbol{x} \in \mathcal{X} |\ d(\boldsymbol{x}, \boldsymbol{w}^i) < d(\boldsymbol{x}, \boldsymbol{w}^j), \forall i \neq j\} \ . \tag{2.4}$$

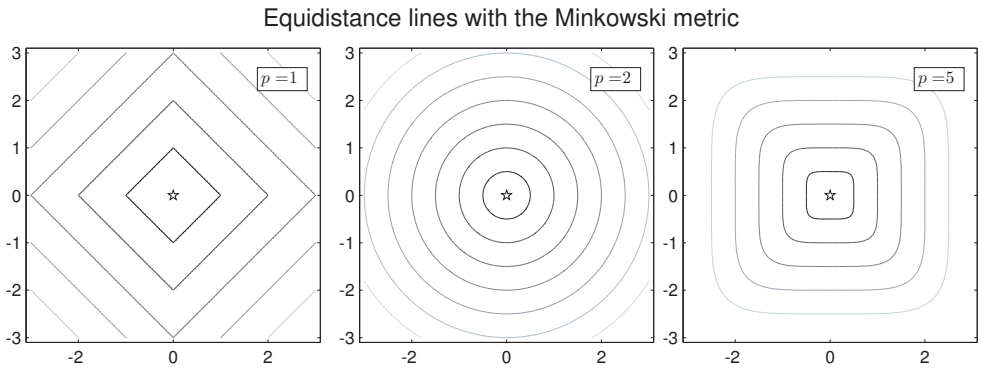


Figure 2.1: Visualization of the equidistance lines from the origin using the Minkowski metric with different values of p.

Figure 2.2 shows two examples of nearest prototype classification on a three class problem using different distance measures. The Euclidean distance leads piecewise linear decision boundaries and receptive fields. For different values of p in the Minkowski metric more general decision boundaries can be realized.

The number of protoytpes is a hyper-parameter of the model and has to be optimized by means of a validation procedure. Too few prototypes may not represent the data structure sufficiently, which yields poor classification performance and too many prototypes may cause overfitting leading to poor generalization ability of the classifier. Many machine learning techniques have been proposed based on the nearest prototype classification scheme. Some of them used in the thesis will be addressed in the next sections.

2.3 Generalized Learning Vector Quantization

Generalized LVQ (GLVQ) (Sato and Yamada 1996) was proposed as a variant of the original LVQ algorithms (Kohonen 1986) derived from an explicit cost function. The method is designed as online-learning algorithm, i.e. the training samples are presented iteratively in each iteration i causing a parameter update only dependent on the current example $(\boldsymbol{x}^i, y^i)$. The aim is to place the prototypes $\boldsymbol{w}^j$ such that a high classification accuracy on novel data after training is achieved. Assuming training data $\{(\boldsymbol{x}^i, y^i)\}_{i=1}^n$ the cost function is defined by

$$E_{\mathrm{GLVQ}} = \sum_{i=1}^{n} \Phi(\mu^i), \quad \text{with} \quad \mu^i = \frac{d^J - d^K}{d^J + d^K} \ , \tag{2.5}$$

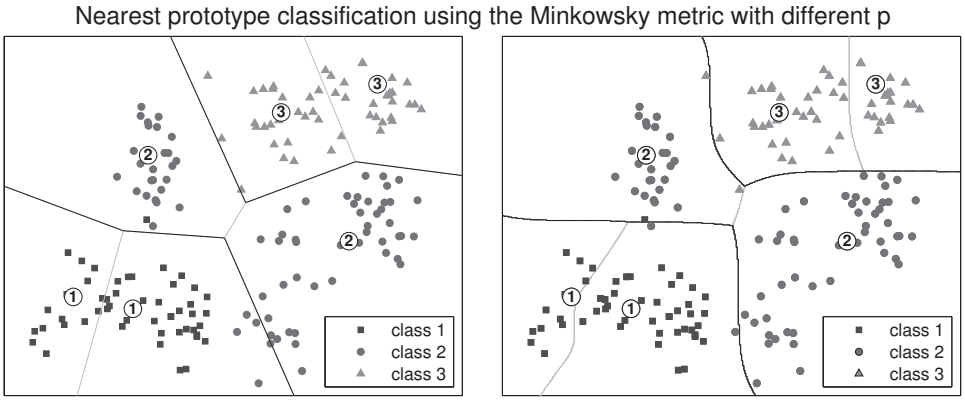


Figure 2.2: Visualization of the decision bounds of a nearest prototype classification scheme using different distances. The data is consisting of 3 classes and each class is represented by two prototypes. The Euclidean distance (left panel) shows piecewise linear boundaries where the gray lines denote the receptive fields of each prototype. In the right panel the Minkowski metric of order $p = 5$ is used.

with $d^J = d(\boldsymbol{x}^i, \boldsymbol{w}^J)$ and $d^K = d(\boldsymbol{x}^i, \boldsymbol{w}^K)$ denote the squared Euclidean distance of the closest prototype with the same and a different class label compared to the actual sample $\boldsymbol{x}^i$ respectively. Φ is a monotonically increasing function, such as a sigmoidal function $\Phi(t) = (1 + \exp(-t))^{-1}$ or the identity $\Phi(t) = t$. The relative difference distance μ can be interpreted as a measure of confidence of the classification. A negative numerator indicates a correct classification. The smaller the value of the numerator the larger the distance of the closest wrong prototype and the bigger the security of the classifiers decision. With the denominator μ is scaled to the interval $[-1, 1]$. The cost function is heuristically motivated. Nevertheless, it has been shown, that it corresponds to large margin optimization, so that good generalization ability is expected (Hammer, Strickert and Villmann 2005a).

In GLVQ the learning rules are given following a steepest descent procedure to minimize the costs. It can be shown that GLVQ is a generalized model that, with respective choice for Φ and μ, includes the conventional LVQ schemes, such as LVQ1 and LVQ2.1 (Sato and Yamada 1996). The learning rules derived from Eq. (2.5) are similar to LVQ2.1:

$$\boldsymbol{w}^J \leftarrow \boldsymbol{w}^J + \tau \frac{\partial \Phi(\mu^i)}{\partial \mu^i} \frac{2d^K}{(d^J + d^K)^2} \cdot (\boldsymbol{x}^i - \boldsymbol{w}^J) \tag{2.6}$$

$$\boldsymbol{w}^K \leftarrow \boldsymbol{w}^K - \tau \frac{\partial \Phi(\mu^i)}{\partial \mu^i} \frac{2d^J}{(d^J + d^K)^2} \cdot (\boldsymbol{x}^i - \boldsymbol{w}^K) \ , \tag{2.7}$$

where $\tau > 0$ is the learning rate or update strength. The closest correct prototype $\boldsymbol{w}^J$ is attracted by the current training sample, while the closest incorrect prototype $\boldsymbol{w}^K$ is repelled. The learning rule (2.6) and (2.7) with sigmoidal Φ shows particular powerful and noise tolerant behavior since it combines adaptation near the optimum Bayesian borders like LVQ2.1, while prohibiting the possible divergence of LVQ2.1 as reported in (Sato and Yamada 1996). The cost function Eq. (2.5) is non-convex, so, as for the other LVQ variants, the learning dynamics depend on the initial state of the system and may suffer from local minima. Often the prototypes are initialized near the class conditional means. The learning is performed until a stopping criterion is fulfilled, e.g. convergence or the maximal number of iterations is reached. One sweep through the complete training set is referred to as an *epoch*. A short description of the algorithm is given in Algorithm 2.1.

Algorithm 2.1 : Generalized LVQ (GLVQ)

1: initialize the prototypes $\boldsymbol{w}^j$
2: **while** stopping criterion not reached **do**
3: randomly select a training sample $\boldsymbol{x}^i$
4: determine closest correct prototype $\boldsymbol{w}^J = \arg\min_j d(\boldsymbol{x}^i, \boldsymbol{w}^j)$ with $y^i = c(\boldsymbol{w}^J)$ and the closest incorrect prototype $\boldsymbol{w}^K = \arg\min_j d(\boldsymbol{x}^i, \boldsymbol{w}^j)$ with $y^i \neq c(\boldsymbol{w}^K)$
5: update the prototypes according to Eq. (2.6) and (2.7)
6: **end while**

2.4 Adaptive metrics in Learning Vector Quantization

The classification schemes mentioned before crucially depend on the dissimilarity measure used. A common choice is the (squared) Euclidean distance, which evaluates the similarity of two feature vectors by equally weighted input dimensions: i.e. equidistance points lie on a hypersphere around the target point. This might be inappropriate for data sets in which features are correlated or not equally scaled. Furthermore, noisy dimensions contribute equally to the computation of the distance and may impair the classification accuracy. Therefore data has to be preprocessed and scaled appropriately, such that the input dimensions have approximately the same importance for classification.

Metric adaptation techniques have been investigated to overcome some problems mentioned before. The aim is to learn a discriminative distance from training data optimized for the specific application. Early proposals introduce weighting factors λ_i to the data dimensions x_i which are automatically adapted (Bojer et al. 2001).

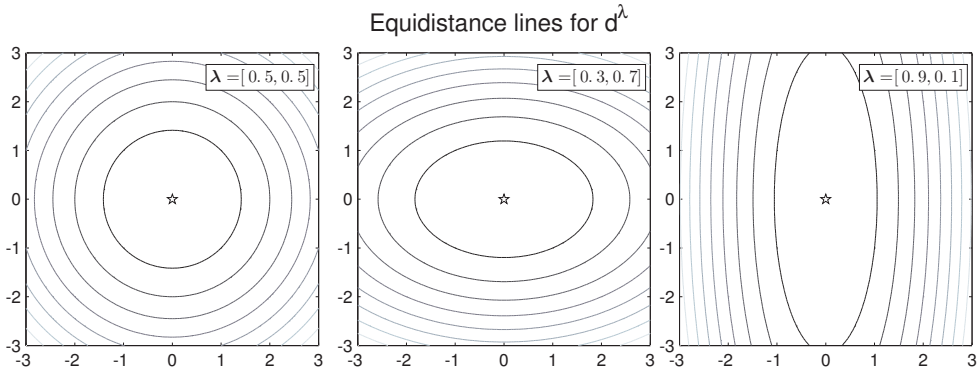


Figure 2.3: Visualization of equidistance lines from the origin using the scaled Euclidean distance d^{λ} with different relevances $\boldsymbol{\lambda}$. The left panel shows the case similar to the Euclidean distance.

They substituted the squared Euclidean distance by a parameterized dissimilarity incorporating a relevance vector $\boldsymbol{\lambda}$ with $\lambda_i \geqslant 0$ and $\sum_{i=1}^{N} \lambda_i = 1$. The adaptive vector introduces a weight for each input dimension:

$$d^{\lambda}(\boldsymbol{x}, \boldsymbol{w}) = \sum_{i=1}^{N} \lambda_i (x_i - w_i)^2 \quad . \tag{2.8}$$

These weights can be interpreted as importance of the respective feature for the classification: weights of noisy, redundant or non-informative dimensions are reduced, while discriminative features gain higher values. The illustration of equidistance lines using d^{λ} is depicted in Figure 2.3. A Hebbian learning step was added to the original LVQ1 learning rule, which updates the relevance vector $\boldsymbol{\lambda}$ in each iteration. The new algorithm was called Relevance LVQ (RLVQ) (Bojer et al. 2001). The Hebbian learning step inherited from LVQ1 showed some instabilities for large data sets, which are subject to noise, hence, the GLVQ (sec. 2.3) was extended with respect to the adaptive metric Eq. (2.8) (Hammer and Villmann 2002). The resulting algorithm is called Generalized Relevance LVQ (GRLVQ). The relevance update is given by the derivative of the cost function Eq. (2.5) with respect to $\boldsymbol{\lambda}$ and reads

$$\lambda_m \leftarrow \lambda_m - \epsilon \cdot \frac{\partial \Phi(\mu^i)}{\partial \mu^i} \left(\frac{d^K}{(d^J + d^K)^2} (x_m^i - w_m^J)^2 - \frac{d^J}{(d^J + d^K)^2} (x_m^i - w_m^K)^2 \right) \quad , \tag{2.9}$$

with $d^J = d^{\lambda}(\boldsymbol{x}^i, \boldsymbol{w}^J)$ and $d^K = d^{\lambda}(\boldsymbol{x}^i, \boldsymbol{w}^K)$ computed using the scaled distance Eq. (2.8). The pseudocode for GRLVQ is depicted in Algorithm 2.2.

Relevances of features might change within the data space. Localized GRLVQ

Algorithm 2.2 : Generalized Relevance LVQ (GRLVQ)

1: initialize the prototypes $\boldsymbol{w}^j$
2: initialize relevance vector $\boldsymbol{\lambda}$
3: **while** stopping criterion not reached **do**
4: randomly select a training sample $\boldsymbol{x}^i$
5: compute the distances $d^j = d^\lambda(\boldsymbol{x}^i, \boldsymbol{w}^j)$ to the prototypes $\boldsymbol{w}^j$
6: determine closest correct $\boldsymbol{w}^J = \arg\min_j d^\lambda(\boldsymbol{x}^i, \boldsymbol{w}^j)$ with $y^i = c(\boldsymbol{w}^J)$
and closest incorrect $\boldsymbol{w}^K = \arg\min_j d^\lambda(\boldsymbol{x}^i, \boldsymbol{w}^j)$ with $y^i \neq c(\boldsymbol{w}^K)$
7: update the prototypes according to Eq. (2.6) and (2.7)
8: update the relevances according to Eq. (2.9)
9: **end while**

(LGRLVQ) addresses this by localized relevance factors attached to each prototype:

$$d^{\lambda^j}(\boldsymbol{x}, \boldsymbol{w}^j) = \sum_{i=1}^{N} \lambda_i^j (x_i - w_i^j)^2 \tag{2.10}$$

causing an individual update for $\boldsymbol{\lambda}^L$ together with their corresponding prototypes $\boldsymbol{w}^L$ for $L \in \{J, K\}$ (Hammer, Schleif and Villmann 2005).

Relevance learning in LVQ has shown to improve not only the classification performance, but also enhance the interpretability of the model. The relevance profile can directly be interpreted as the contribution of the dimensions for the classification problem and can be used to find suitable candidate features for pruning to save costly measurements. This has turned out particularly suitable in many practical applications containing irrelevant or inadequately scaled dimensions (Mendenhall and Merényi 2006, Biehl et al. 2007, Kietzmann et al. 2008). Further, the generalization ability have been investigated in (Hammer, Strickert and Villmann 2005a). It has been shown, that for an adaptive diagonal metric $\Lambda = \mathrm{diag}(\boldsymbol{\lambda})$, large margin generalization bounds can be derived independent from the dimensionality.

The principle of feature weighting can be developed further by taking into account pairwise correlations of features. Recently, an LVQ extension called Generalized Matrix LVQ (GMLVQ) was proposed (Schneider et al. 2009b, Schneider et al. 2009a). It uses an adaptive metric of the form of a Mahalanobis distance

$$d^\Lambda(\boldsymbol{w}, \boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{w})^\top \Lambda (\boldsymbol{x} - \boldsymbol{w}) \tag{2.11}$$

with a parameter matrix $\Lambda \in \mathbb{R}^{N \times N}$. The matrix Λ is assumed to be positive (semi-) definite and we can substitute

$$\Lambda = \Omega^\top \Omega \quad \text{with } \Omega \in \mathbb{R}^{N \times N} \ . \tag{2.12}$$

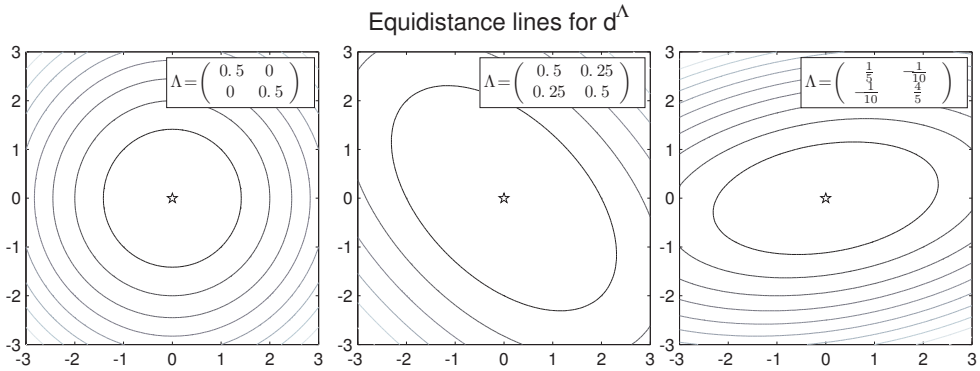


Figure 2.4: Visualization of equidistance lines from the origin using the adaptive distance d^Λ with different relevance matrices Λ. The left panel shows a metric similar to the Euclidean distance.

Hence, the measure corresponds to a (squared) Euclidean distance in an appropriately transformed space

$$d^\Lambda(\boldsymbol{x}, \boldsymbol{w}) = \left[\Omega\ (\boldsymbol{x} - \boldsymbol{w})\right]^2 \tag{2.13}$$

with an arbitrary matrix Ω. Specific restrictions may be imposed on Ω without loss of generality. Note that, for instance, every positive symmetric Λ has a symmetric root Ω with $\Lambda = \Omega^2$. The equidistance lines for some example configurations of d^Λ are visualized in Figure 2.4. Using relevance matrices allows to detect alternative directions in the feature space and therefore provide more discriminative power to separate the classes. Also GMLVQ was introduced as an extension of GLVQ and therefore inherits the same cost function, substituting the squared Euclidean distance in the original formulation Eq. (2.5) by the adaptive metric:

$$E_{\mathrm{GMLVQ}} = \sum_i \Phi(\mu^i), \quad \text{with} \quad \mu^i = \frac{d_J^\Lambda - d_K^\Lambda}{d_J^\Lambda + d_K^\Lambda}\ . \tag{2.14}$$

The quantities $d_J^\Lambda = d^\Lambda(\boldsymbol{x}^i, \boldsymbol{w}^J)$ and $d_K^\Lambda = d^\Lambda(\boldsymbol{x}^i, \boldsymbol{w}^K)$ correspond again to the distances of the actual feature vector $\boldsymbol{x}^i$ from the closest correct prototype $\boldsymbol{w}^J$ and the closest incorrect prototype $\boldsymbol{w}^K$, respectively. The original GMLVQ algorithm corresponds to a stochastic gradient descent in the cost function, Eq. (2.14), with respect to the prototype configuration and an arbitrary matrix Ω. Gradients are evaluated with respect to the contribution of single instances $\boldsymbol{x}^i$, which are presented random sequentially. The GMLVQ method is summarized in Algorithm 2.3:

Algorithm 2.3 : Generalized Matrix LVQ (GMLVQ)

1: initialize the prototypes $\boldsymbol{w}^j$
2: initialize matrix Ω and normalize according to Eq. (2.21)
3: **while** stopping criterion not reached **do**
4: randomly select a training sample $\boldsymbol{x}^i$
5: compute the distances $d^\Lambda(\boldsymbol{x}^i, \boldsymbol{w}^j)$ to the prototypes $\boldsymbol{w}^j$
6: determine closest correct $\boldsymbol{w}^J = \arg\min_j d^\Lambda(\boldsymbol{x}^i, \boldsymbol{w}^j)$ with $y^i = c(\boldsymbol{w}^J)$
and closest incorrect $\boldsymbol{w}^K = \arg\min_j d^\Lambda(\boldsymbol{x}^i, \boldsymbol{w}^j)$ with $y^i \neq c(\boldsymbol{w}^K)$
7: update the prototypes according to $\boldsymbol{w}^L \leftarrow \boldsymbol{w}^L - \tau_1 \cdot \frac{\partial E_{\mathrm{GMLVQ}}}{\partial \boldsymbol{w}^L}$, $L \in \{J, K\}$
8: update the matrix according to $\Omega \leftarrow \Omega - \tau_2 \cdot \frac{\partial E_{\mathrm{GMLVQ}}}{\partial \Omega}$
9: normalize the matrix according to Eq. (2.21)
10: **end while**

The derivative of E_{GMLVQ} with respect to the prototypes is given by:

$$\frac{\partial E_{\mathrm{GMLVQ}}}{\partial \boldsymbol{w}^L} = \frac{\Phi(\mu^i)}{\partial \mu^i} \cdot \frac{\partial \mu^i}{\partial d_L^\Lambda} \cdot \frac{\partial d_L^\Lambda}{\partial \boldsymbol{w}^L} = \Phi' \cdot \gamma^L \cdot \frac{\partial d_L^\Lambda}{\partial \boldsymbol{w}^L} \quad \text{where } L \in \{J, K\} \tag{2.15}$$

$$\text{with } \gamma^J = \frac{\partial \mu}{\partial d_J^\Lambda} = \frac{2 d_K^\Lambda}{(d_J^\Lambda + d_K^\Lambda)^2} \ , \tag{2.16}$$

$$\gamma^K = \frac{\partial \mu}{\partial d_K^\Lambda} = \frac{-2 d_J^\Lambda}{(d_J^\Lambda + d_K^\Lambda)^2} \ , \tag{2.17}$$

$$\text{and } \frac{\partial d_L^\Lambda}{\partial \boldsymbol{w}^L} = -2 \Omega^\top \Omega (\boldsymbol{x}^i - \boldsymbol{w}^L) \ . \tag{2.18}$$

The derivatives corresponding to the elements of Ω_{mn} read:

$$\frac{\partial E_{\mathrm{GMLVQ}}}{\partial \Omega_{mn}} = \frac{\Phi(\mu^i)}{\partial \mu^i} \cdot \frac{\partial \mu^i}{\partial \Omega_{mn}} = \Phi' \cdot \left(\gamma^J \frac{\partial d_J^\Lambda}{\partial \Omega_{mn}} + \gamma^K \frac{\partial d_K^\Lambda}{\partial \Omega_{mn}} \right) \ , \tag{2.19}$$

$$\frac{\partial d_L^\Lambda}{\partial \Omega_{mn}} = 2 \sum_j^N (x_n^i - w_n^L) \Omega_{mj} (x_j^i - w_j^L) = 2 \left[\Omega (\boldsymbol{x}^i - \boldsymbol{w}^L) \right]_m (x_n^i - w_n^L). \tag{2.20}$$

After each learning step the matrix Λ is normalized to prevent the algorithm from degeneration. One possibility is to enforce

$$\sum_i \Lambda_{ii} = \sum_{ik} \Omega_{ki} \Omega_{ki} = \sum_{ik} (\Omega_{ki})^2 = 1 \tag{2.21}$$

by dividing all elements of Ω by $\left(\sum_{ki} (\Omega_{ki})^2 \right)^{\frac{1}{2}}$. The sum of diagonal elements $\sum_i \Lambda_{ii}$ coincides with the sum of eigenvalues. This generalizes the normalization of relevances $\sum_i \lambda_i = 1$ for a simple diagonal metric.

Alternatively, similar to the LGRLVQ scheme Eq. (2.10), local matrices Λ^j can be attached to every prototype or to the prototypes of each class (Schneider et al. 2009b, Schneider et al. 2009a). The corresponding dissimilarity measure

$$d^{\Lambda^j}(\boldsymbol{x}, \boldsymbol{w}^j) = (\boldsymbol{x} - \boldsymbol{w}^j)^\top \Lambda^j (\boldsymbol{x} - \boldsymbol{w}^j) \text{ with } \Lambda^j = \Omega^{j\top}\Omega^j \tag{2.22}$$

has the potential to take into account correlations varying between different classes or regions of the feature space. Thus, clusters with ellipsoidal shape and different orientations can be presented in the data. The cost function of this Localized GMLVQ (LGMLVQ) is defined including the localized distances $d_J^{\Lambda^J} = d^{\Lambda^J}(\boldsymbol{x}^i, \boldsymbol{w}^J)$ and $d_K^{\Lambda^K} = d^{\Lambda^K}(\boldsymbol{x}^i, \boldsymbol{w}^K)$, with the indices J and K again referencing the closest correct and incorrect prototype respectively:

$$E_{\mathrm{LGMLVQ}} = \sum_i \Phi(\mu^i_{\mathrm{local}}), \quad \text{with} \quad \mu^i_{\mathrm{local}} = \frac{d_J^{\Lambda^J} - d_K^{\Lambda^K}}{d_J^{\Lambda^J} + d_K^{\Lambda^K}} \ . \tag{2.23}$$

The LGMLVQ is depicted in Algorithm 2.4.

Algorithm 2.4 : Localized GMLVQ (LGMLVQ)

1: initialize the prototypes $\boldsymbol{w}^j$
2: initialize matrices Ω^j and normalize according to Eq. (2.21)
3: **while** stopping criterion not reached **do**
4: randomly select a training sample $\boldsymbol{x}^i$
5: compute the distances $d^{\Lambda^j}(\boldsymbol{x}^i, \boldsymbol{w}^j)$ to the prototypes $\boldsymbol{w}^j$
6: determine closest correct $\boldsymbol{w}^J = \arg\min_j d^{\Lambda^j}(\boldsymbol{x}^i, \boldsymbol{w}^j)$ with $y^i = c(\boldsymbol{w}^J)$
and closest incorrect $\boldsymbol{w}^K = \arg\min_j d^{\Lambda^j}(\boldsymbol{x}^i, \boldsymbol{w}^j)$ with $y^i \neq c(\boldsymbol{w}^K)$
7: update the prototypes according to $\boldsymbol{w}^L \leftarrow \boldsymbol{w}^L - \tau_1 \cdot \frac{\partial E_{\mathrm{LGMLVQ}}}{\partial \boldsymbol{w}^L}$, $L \in \{J, K\}$
8: update the matrices according to $\Omega^L \leftarrow \Omega^L - \tau_2 \cdot \frac{\partial E_{\mathrm{LGMLVQ}}}{\partial \Omega^L}$
9: normalize the matrices according to Eq. (2.21)
10: **end while**

The derivative of E_{LGMLVQ} with respect to the prototypes is given by:

$$\frac{\partial E_{\mathrm{LGMLVQ}}}{\partial \boldsymbol{w}^L} = \frac{\Phi(\mu^i_{\mathrm{local}})}{\partial \mu^i_{\mathrm{local}}} \cdot \frac{\partial \mu^i_{\mathrm{local}}}{\partial d_L^{\Lambda^L}} \cdot \frac{\partial d_L^{\Lambda^L}}{\partial \boldsymbol{w}^L} \quad \text{where } L \in \{J, K\}, \tag{2.24}$$

$$\gamma^J_{\mathrm{local}} = \frac{\partial \mu_{\mathrm{local}}}{\partial d_J^{\Lambda^J}} = \frac{2 d_K^{\Lambda^K}}{(d_J^{\Lambda^J} + d_K^{\Lambda^K})^2} \ , \tag{2.25}$$

$$\gamma^K_{\mathrm{local}} = \frac{\partial \mu_{\mathrm{local}}}{\partial d_K^{\Lambda^K}} = \frac{-2 d_J^{\Lambda^J}}{(d_J^{\Lambda^J} + d_K^{\Lambda^K})^2} \ , \tag{2.26}$$

and

$$\frac{\partial d_L^{\Lambda^L}}{\partial \boldsymbol{w}^L} = -2\Omega^{L\top}\Omega^L(\boldsymbol{x}^i - \boldsymbol{w}^L) \ . \tag{2.27}$$

The derivatives corresponding to the elements of Ω_{mn} read:

$$\frac{\partial E_{\mathrm{LGMLVQ}}}{\partial \Omega_{mn}^L} = \frac{\Phi(\mu^i_{\mathrm{local}})}{\partial \mu^i_{\mathrm{local}}} \cdot \frac{\partial \mu^i_{\mathrm{local}}}{\partial \Omega^L_{mn}} = \Phi' \cdot \gamma^L_{\mathrm{local}} \cdot \frac{\partial d_L^{\Lambda^L}}{\partial \Omega^L_{mn}} \text{ with } L \in \{J, K\} \tag{2.28}$$

$$\frac{\partial d_L^{\Lambda^L}}{\partial \Omega^L_{mn}} = 2\sum_j^N (x^i_n - w^L_n)\Omega^L_{mj}(x^i_j - w^L_j) = 2\left[\Omega^L(\boldsymbol{x}^i - \boldsymbol{w}^L)\right]_m (x^i_n - w^L_n). \tag{2.29}$$

Local matrices increase the capacity of the system by implying nonlinear decision boundaries. The receptive fields of the prototypes need no longer be convex or even connected. Example visualizations of global and local matrices are shown in Chapter 3 and Part II of the thesis.

2.5 Large Margin Nearest Neighbor

The k-NN algorithm is a simple and intuitive method which classifies a novel feature vector by a majority vote among its k nearest neighbors in the training set. Thus, its performance depends crucially on the metric used for the identification of the neighbors. The Large Margin Nearest Neighbor (LMNN) (Weinberger et al. 2006) algorithm extends the k-NN rule by an adaptive distance measure. The aim of the training process is that a predefined number κ of nearest neighbors (called target neighbors) belongs to the same class like the example data with high probability. Simultaneously, samples of different classes should be separated by a large margin. Figure 2.5 illustrates this concept. Therefor, the LMNN algorithm provides a discriminative distance measure for the k-NN classifier corresponding to

$$d^\Gamma(\boldsymbol{x}^i, \boldsymbol{x}^j) = (\boldsymbol{x}^i - \boldsymbol{x}^j)^\top \Gamma (\boldsymbol{x}^i - \boldsymbol{x}^j) \ , \tag{2.30}$$

where the matrix $\Gamma \in \mathbb{R}^{N\times N}$ denotes the counterpart of Λ used in GMLVQ.

The training procedure has two steps. The first step identifies a set of κ similarly labeled target neighbors for each input $\boldsymbol{x}^i$. Whereby, the computational effort depends crucially on the parameter κ. The second step adapts the Mahalanobis distance metric such that these target neighbors are closer to $\boldsymbol{x}^i$ than differently labeled inputs. The semi-definite optimization in LMNN classification arises from an objective function:

$$E = (1-b)\sum_{i,j \rightsquigarrow i} d^\Gamma(\boldsymbol{x}^i, \boldsymbol{x}^j) + b \sum_{i,j \rightsquigarrow i,l} \left(1 - Y^{il}\right)\left[1 + d^\Gamma(\boldsymbol{x}^i, \boldsymbol{x}^j) - d^\Gamma(\boldsymbol{x}^i, \boldsymbol{x}^l)\right]_+, \tag{2.31}$$

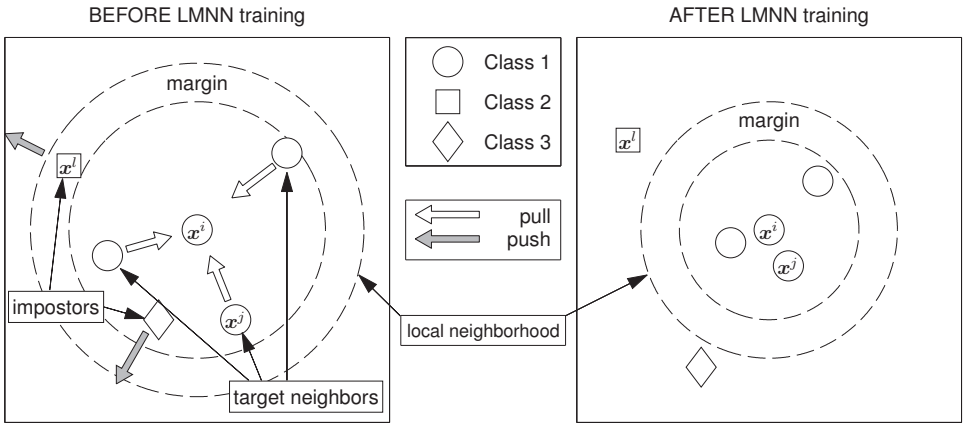


Figure 2.5: Illustration of the neighborhood before and after LMNN training.

where $[z]_+ = \max(z, 0)$ denotes the standard hinge loss. The constant b defines the trade-off between the two terms: the first part penalizes large distances between inputs and their target neighbors, while the second part penalizes small distances between differently labeled inputs.

The terms in the objective function can be specified with following notation: $Y^{ij} \in \{0, 1\}$ indicate whether the inputs $\boldsymbol{x}^i$ and $\boldsymbol{x}^j$ have the same class label. The notation $j \rightsquigarrow i$ indicates that $\boldsymbol{x}^j$ is a target neighbor of $\boldsymbol{x}^i$. Also, let the slack variables $X^{ijl} \geqslant 0$ denote the amount by which a differently labeled input $\boldsymbol{x}^l$ (impostor) invades the perimeter around input $\boldsymbol{x}^i$ and its target neighbors $\boldsymbol{x}^j$. The matrix Γ in the quadratic form Eq. (2.30) is obtained by solving the semidefinite program shown in Algorithm 2.5.

Algorithm 2.5 : Semidefinite optimization problem in LMNN

Minimize $(1-b)\sum_{i,j \rightsquigarrow i} d^{\Gamma}(\boldsymbol{x}^i, \boldsymbol{x}^j) + b\sum_{i,j \rightsquigarrow i,l}(1 - Y^{il})X^{ijl}$ **subject to:**

(a) $d^{\Gamma}(\boldsymbol{x}^i, \boldsymbol{x}^l) - d^{\Gamma}(\boldsymbol{x}^i, \boldsymbol{x}^j) \geqslant 1 - X^{ijl}$

(b) $X^{ijl} \geqslant 0$

(c) $\Gamma \succeq 0$

The constraints of type (a) favor inputs $\boldsymbol{x}^i$ closer to their κ target neighbors $\boldsymbol{x}^j$ then to any other differently labeled input $\boldsymbol{x}^l$. When differently labeled $\boldsymbol{x}^l$ invade the local neighborhood a positive slack variable X^{ijl} is generated. This is penalized in the second term of the objective function. Constraints of type (b) enforce non-negativity of the slack variables and constraint (c) enforces positive semi-

definiteness of Γ. Noting that the quadratic form d^{Γ} is linear in the matrix Γ, the above optimization is easily recognized as a semidefinite problem. MATLAB code[1] of the algorithm is provided and used for the experiments in this thesis.

[1] www.cse.wustl.edu/∼kilian/code/code.html

Published as:

K. Bunte, P. Schneider, B. Hammer, F.-M. Schleif, T. Villmann and M. Biehl – *"Discriminative Visualization by Limited Rank Matrix Learning,"* Leipzig University, Machine Learning Reports (2:3), pp. 37–51, 2008.

K. Bunte, P. Schneider, B. Hammer, F.-M. Schleif, T. Villmann and M. Biehl – *"Limited Rank Matrix Learning Discriminative Dimension Reduction and Visualization,"* accepted for publication in Neural Networks 2011.

Chapter 3

Limited Rank Matrix LVQ

Projection makes it possible.

The impossible triangle. Shigeo Fukuda

Abstract

We present an extension of the Generalized Matrix Learning Vector Quantization algorithm. In the original scheme, adaptive square matrices of relevance factors parameterize a discriminative distance measure. We extend the scheme to matrices of limited rank corresponding to low-dimensional representations of the data. This allows to incorporate prior knowledge of the intrinsic dimension and to reduce the number of adaptive parameters efficiently. In particular, for very high dimensional data, the limitation of the rank can reduce computation time and memory requirements significantly. Furthermore, two- or three-dimensional representations constitute an efficient visualization method for labeled data sets. The identification of a suitable projection is not treated as a pre-processing step but as an integral part of the supervised training. Several real world data sets serve as illustration and demonstrate the usefulness of the suggested method.

3.1 Introduction

In (Schneider et al. 2009b, Schneider et al. 2009a) the concept of GMLVQ is introduced. It uses the quadratic form Eq. (2.11) as distance including a full matrix

of relevances, which can account for correlations between different features. An adaptive self-affine transformation Ω (see Eq. (2.12)) of feature space identifies the coordinate system which is most suitable for the given classification task. The original formulation of GMLVQ employs symmetric squared matrices $\Omega \in \mathbb{R}^{N \times N}$ and is summarized in Algorithm 2.3. In the simplest case, one matrix is taken to define a global distance measure. Extensions to class-wise or local matrices, attached to individual prototypes Eq. (2.22), are technically straightforward and allow for the parameterization of more complex decision boundaries.

In this chapter we present and discuss an important modification: the use of rectangular transformation matrices $\Omega \in \mathbb{R}^{M \times N}$ with $M \leqslant N$ (Bunte et al. 2008, Bunte, Schneider, Hammer, Schleif, Villmann and Biehl 2011). The corresponding relevance matrices Λ are of bounded rank M or, in other words, distances are evaluated in a space with reduced dimension, see Eq. (2.13). The motivation for considering this variation of GMLVQ is at least two-fold: (a) prior knowledge about the intrinsic dimension of the data can be incorporated efficiently and (b) the number of free parameters in the learning problem may be reduced significantly.

Although unrestricted GMLVQ displays a tendency to reduce the rank of the relevance matrices in the training process, the advantages of restricting the rank explicitly are obvious. In particular for nominally very high-dimensional data, e.g. in image analysis or bioinformatics, unrestricted relevance matrices become intractable. In addition, optimization results can be poor when the search is performed in an unnecessarily large parameter space. Furthermore, the exact control of the rank allows for pre-defining the dimension of the intrinsic representation and is, for instance, suitable for the discriminative visualization of labeled data sets. In contrast with many other schemes that consider dimension reduction as a pre-processing step, our method performs the training of prototypes and the identification of a suitable transformation simultaneously. Hence, both sub-tasks are guided by the ultimate goal of implementing the desired classification scheme.

Appropriate projections into two- or three-dimensional spaces can furthermore be used for efficient visualization of labeled data. Visualization enables to use the astonishing cognitive capabilities of humans for visual perception when extracting information from large data volumes. Structural characteristics can be captured almost instantly by humans, independent of the number of displayed points. Classical unsupervised dimension reduction techniques represent data points contained in a high dimensional data manifold by low dimensional counterparts in, for instance, two or three dimensions, while preserving as much information as possible. Since it is not clear in advance which parts of the data are relevant to the user, this problem is inherently ill-posed: depending on the specific data domain and the situation at hand, different aspects can be in the focus of attention. Prior knowledge, in form of

label information, can be used to formulate a well-defined objective in terms of the classification performance.

There exist a few classical dimensionality reduction tools which take class labels into account: e.g. Classical Fisher Linear Discriminant Analysis (LDA), the recently introduced local Fisher discriminant analysis (LFDA) (Sugiyama and Roweis 2007), Neighborhood Component Analysis (NCA) (Goldberger et al. 2004), as well as partial Least Squares regression (PLS). These methods can be extended to nonlinear projections by kernel methods (Ma et al. 2007, Baudat and Anouar 2000). Adaptive dissimilarity measures which modify the metric according to the given auxiliary information have been introduced e.g. in (Kaski et al. 2001, Peltonen et al. 2004, Bunte, Hammer, Schneider and Biehl 2009, Bunte, Hammer and Biehl 2009, Bunte, Hammer, Wismüller and Biehl 2010).The resulting metric can be integrated into various techniques such as SOM, Multidimensional Scaling (MDS), or a recent information theoretic model for data visualization (Kaski et al. 2001, Peltonen et al. 2004, Venna et al. 2010). An ad hoc metric adaptation is used in (Geng et al. 2005) to extend Isomap (Tenenbaum et al. 2000) to class labels. Alternative approaches change the cost function of dimensionality reduction, for instance by using conditional probabilities, class-wise similarity matrices or introducing a covariance-based coloring matrix for the side information as proposed in (Iwata et al. 2007, Memisevic and Hinton 2005, Song et al. 2008). The detailed explanation of the most important supervised and unsupervised dimension reduction techniques is given in Part II of this thesis.

In the next section we describe the Limited Rank Matrix LVQ (LiRaM LVQ) as extension of the original GMLVQ formulation. Afterwards we apply the novel approach to a benchmark problem and study the influence of the dimension reduction on the classification performance. We also compare the limited rank version to the naive approach of taking the first components of the full rank GMLVQ. We show that reducing the rank after training not only requires more memory and CPU time, but also yields inferior classification performance compared to LiRaM LVQ. In Sec. 3.4 we present example applications of our algorithm in the visualization of labeled data. We also compare with visualizations obtained by LFDA and NCA. We conclude by summarizing our findings and providing an outlook on perspective investigations.

3.2 Limited Rank Matrix LVQ

In the following we extend the concept of GMLVQ to the use of rectangular matrices in the distance measure and refer to the corresponding algorithm as LiRaM LVQ.

Basically, we follow the same procedure as depicted in Algorithm 2.3 for GMLVQ, but we consider Ω from Eqs. (2.11) and (2.12) to define a transformation from the original N-dimensional feature space to $\mathbb{R}^M$ with $M \leqslant N$ so that:

$$\Lambda = \Omega^\top \Omega \quad \text{with } \Omega \in \mathbb{R}^{M \times N}. \tag{3.1}$$

This section addresses the use of one global matrix for the dimension reduction and visualization. Modifications in the sense of extensions towards local distance measures will be discussed in the next section.

Note that, in general, the transformation matrix Ω is not uniquely determined. The distance measure is, for instance, invariant under rotations in feature space. We can identify a unique $\widehat{\Omega}$ by decomposing $\Lambda = \Omega^\top \Omega$ in a canonical way: We determine the normalized eigenvectors $\boldsymbol{v}^1, \boldsymbol{v}^2, \ldots, \boldsymbol{v}^M$ corresponding to the M ordered non-zero eigenvalues of Λ, $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_M$ and define $\widehat{\Omega}$ as:

$$\widehat{\Omega} = \left(\left[\sqrt{\lambda_1} \boldsymbol{v}^1, \sqrt{\lambda_2} \boldsymbol{v}^2, \ldots, \sqrt{\lambda_M} \boldsymbol{v}^M \right] \right)^\top \tag{3.2}$$

In addition we choose the sign of v_i, such that the component of v_i with largest magnitude is positive. Note, that the value M limits the rank of the dissimilarity matrix Λ to a maximum of M. Nevertheless, the matrix can be forced to keep the given rank by recently introduced regularization schemes (Schneider et al. 2010). With the scheme Eq. (3.2) also a full matrix can be restricted after training. However, if eigenvectors with eigenvalues bigger than zero are omitted classification accuracy might get lost. We discuss this in section 3.3. Nominally, the matrix Ω will have more independent entries than the symmetric Λ whenever $M > (N+1)/2$. However, we have found no evidence that this ambiguity complicates the optimization problem. Therefore we consider throughout the following, general, unrestricted matrices Ω with $M \cdot N$ independent entries.

The update rules for the LiRaM LVQ can be obtained by taking the derivatives of the objective function E_{GMLVQ} Eq. (2.14) with respect to the prototypes $\boldsymbol{w}^L$ with $L \in \{J, K\}$ and the matrix $\Omega \in \mathbb{R}^{M \times N}$. The derivatives are the same as for GMLVQ given in Appendix 3.A and the updates for a given sample $\boldsymbol{x}^i$ read:

$$\boldsymbol{w}^L \leftarrow \boldsymbol{w}^L + \tau_1 \cdot \Phi' \cdot \gamma^L \cdot 2 \cdot \Omega^\top \Omega (\boldsymbol{x}^i - \boldsymbol{w}^L) \text{ with } L \in \{J, K\} \tag{3.3}$$

$$\Omega \leftarrow \Omega - \tau_2 \cdot \Phi' \cdot \left(\gamma^J \frac{\partial d_J^\Lambda}{\partial \Omega_{mn}} + \gamma^K \frac{\partial d_K^\Lambda}{\partial \Omega_{mn}} \right) \tag{3.4}$$

Throughout the thesis we consider the scaling function Φ being the identity $\Phi(a) = a$ with derivative $\Phi' = 1$.

Note that the learning rates τ_1 and τ_2 can be chosen independently. In particular, we set $\tau_1 \gg \tau_2$ which implies that changes of the metric occur on a slower

Algorithm 3.1 : Limited Rank Matrix LVQ (LiRaM LVQ)

Same as Algorithm 2.3 with $\Omega \in \mathbb{R}^{M \times N}$ limiting the rank of Λ to a maximum M

time scale than those of the prototypes. This setting has proven advantageous in many implementations of matrix relevance learning (Bojer et al. 2001, Hammer and Villmann 2002, Schneider et al. 2009b). In all practical examples considered in the following, we apply a learning rate schedule of the form

$$\tau_1(t) = \frac{\tau_1^{\text{start}}}{1 + (t-1)\Delta\tau_1} \quad \text{and} \tag{3.5}$$

$$\tau_2(t) = \begin{cases} \frac{\tau_2^{\text{start}}}{1+(t-t_M)\Delta\tau_2} & \text{for } t \geqslant t_M \\ 0 & \text{for } t < t_M. \end{cases} \tag{3.6}$$

Here, t corresponds to the current epoch, i.e. sweep through the training data set, and $\tau_{1,2}^{\text{start}}$ denotes the initial learning rates. Non-zero relevance updates are performed only after the first t_M epochs of prototype training. The computational costs scale linearly with the number of prototypes $n_{\boldsymbol{w}}$, the dimension of the data N, the target dimension M and with the number of training examples n in each epoch $\mathcal{O}(n_{\boldsymbol{w}}MNn)$.

3.2.1 LiRaM LVQ with localized similarities using two matrices

For full rank matrices the LGMLVQ was introduced in (Schneider et al. 2009b, Schneider et al. 2009a) and is depicted in Algorithm 2.4. It is based on the concept of localized matrices Ω^j in the distance (see Eq. 2.22) individually adapted for each prototype or for each class, flexibly increasing the complexity of the LVQ system. The concept of LiRaM LVQ can also be expanded to the use of localized rectangular matrices, representing several local linear projections. The global combination of these local linear patches by means of charting is discussed in (Brand 2002, Bunte, Hammer, Wismüller and Biehl 2010) and will be discussed in Part II of this thesis.

In this chapter, we will investigate the use of localized matrices in combination with global linear dimension reduction. This can be achieved by expanding the definition of the dissimilarity measure Eq. (2.22) with the combination of two matrices:

$$d_L^{\Psi^L}(\boldsymbol{x}, \boldsymbol{w}^L) = (\boldsymbol{x} - \boldsymbol{w}^L)^\top \Omega^\top \Psi^{L\top} \Psi^L \Omega \, (\boldsymbol{x} - \boldsymbol{w}^L). \tag{3.7}$$

Here $\Omega \in \mathbb{R}^{M \times N}$ performs the dimension reduction with target dimension M, while the $\Psi^L \in \mathbb{R}^{M \times M}$ locally attached to the prototypes $\boldsymbol{w}^L$ define a local dissimilarity measure in the transformed space. Consequently the visualizations show nonlinear

rather than piecewise linear decision boundaries in the M-dimensional space. In the experiments we used class-wise dissimilarities Ψ^c with $c \in \{1, \ldots, C\}$ attached to the prototypes $\boldsymbol{w}^L$ with equal class label $c(\boldsymbol{w}_L) = c$, which may be interesting in a setting with more than one prototype per class. In the following we will address this algorithm as Localized LiRaM LVQ (LLiRaM LVQ).

The update rules for the algorithm are obtained by taking the derivatives of E_{GMLVQ} with respect to the prototypes $\boldsymbol{w}^L$, the transformation $\Omega \in \mathbb{R}^{M \times N}$ and the localized matrices $\Psi^L \in \mathbb{R}^{M \times M}$ with $L \in \{J, K\}$ (see Appendix 3.B). The updates can be summarized by:

$$\boldsymbol{w}^L \leftarrow \boldsymbol{w}^L + \tau_1 \cdot \Phi' \cdot \gamma_\Psi^L \cdot 2\Omega^\top \Psi^{L\top} \Psi^L \Omega \, (\boldsymbol{x}^i - \boldsymbol{w}^L) \text{ with } L \in \{J, K\} \tag{3.8}$$

$$\Omega \leftarrow \Omega - \tau_2 \cdot \Phi' \cdot \left(\gamma_\Psi^J \cdot \frac{\partial d_J^{\Psi^J}}{\partial \Omega} + \gamma_\Psi^K \cdot \frac{\partial d_K^{\Psi^K}}{\partial \Omega} \right) \tag{3.9}$$

$$\Psi^L \leftarrow \Psi^L - \tau_2 \cdot \Phi' \cdot \gamma_\Psi^L \cdot 2 \cdot \Psi^L \left(\Omega(\boldsymbol{x}^i - \boldsymbol{w}^L)(\boldsymbol{x}^i - \boldsymbol{w}^L)^\top \right) \Omega^\top \tag{3.10}$$

The LLiRaM LVQ is depicted in Algorithm 3.2:

Algorithm 3.2 : Localized LiRaM LVQ (LLiRaM LVQ)

initialize the prototypes $\boldsymbol{w}^j$
initialize matrix Ω and normalize according to Eq. (2.21)
initialize matrices Ψ^j
while stopping criterion not reached **do**
  randomly select a training sample $\boldsymbol{x}^i$
  compute the distances $d_j^{\Psi^j}(\boldsymbol{x}^i, \boldsymbol{w}^j)$ to the prototypes $\boldsymbol{w}^j$
  determine closest correct $\boldsymbol{w}^J = \arg\min_j d_j^{\Psi^j}(\boldsymbol{x}^i, \boldsymbol{w}^j)$ with $y^i = c(\boldsymbol{w}^J)$
    and closest incorrect $\boldsymbol{w}^K = \arg\min_j d_j^{\Psi^j}(\boldsymbol{x}^i, \boldsymbol{w}^j)$ with $y^i \neq c(\boldsymbol{w}^K)$
  consider $L \in \{J, K\}$
  update the prototypes according to $\boldsymbol{w}^L \leftarrow \boldsymbol{w}^L - \tau_1 \cdot \frac{\partial E_{\mathrm{GMLVQ}}}{\partial \boldsymbol{w}^L}$ (Eq. (3.8))
  update the matrix $\Omega \leftarrow \Omega - \tau_2 \cdot \frac{\partial E_{\mathrm{GMLVQ}}}{\partial \Omega}$ (Eq. (3.9))
  update the matrices according to $\Psi^L \leftarrow \Psi^L - \tau_2 \cdot \frac{\partial E_{\mathrm{GMLVQ}}}{\partial \Psi^L}$ (Eq. (3.10))
  normalize Ω according to Eq. (2.21)
end while

3.3 A classification problem

As an illustrative example, we study the performance of the LiRaM LVQ algorithm on the image segmentation data set as provided in the UCI repository (Asuncion

et al. 1998). It contains 19-dimensional feature vectors, which have been constructed from regions of 3×3 pixels, randomly drawn from a set of 7 manually segmented outdoor images. The features encode various attributes of the example patches, which have to be assigned to one of the following 7 classes: brickface, sky, foliage, cement, window, path, and grass. The provided data set consists of 210 feature vectors for training, with 30 instances per class. The test set comprises 300 instances per class, i.e. 2100 samples in total. We refer the reader to (Asuncion et al. 1998) for the details. In the data as provided the features 3, 4 and 5 (region-pixel-count, short-line-density-5 and short-line-density-2) display zero variance. Hence, we omit these features and consider only the remaining 16 features. After a z-transformation, each feature displays zero mean and unit variance in the data set.

We apply in the following the LiRaM LVQ algorithm with global matrix Λ and parameters $\tau_1^{start} = 0.01$, $\Delta\tau_1 = 0.0001$, $\tau_2^{start} = 0.001$, $\Delta\tau_2 = 0.0001$ in the schedule Eqs. (3.5) and (3.6), matrix adaptation begins in epoch $t_M = 100$. Similar settings have proven successful in previous applications of the original GMLVQ algorithm to the data set (Schneider et al. 2009a).

3.3.1 Performance dependence on M

We first study the simplest GMLVQ classifiers with only one prototype per class. For several values of M, we perform LiRaM LVQ on the given training set of 210 example data and observe the evolution of training and test accuracies with the number of epochs. In order to obtain reliable results and as an indication of the robustness and convergence properties we present averages and standard deviations with respect to 10 different random initializations of the prototypes and matrix Ω.

Fig. 3.1 shows averaged learning curves for the example cases $M = 2$ and $M = 16$. We display the training and test accuracies averaged over 10 random initializations of the algorithm and the estimates of the corresponding standard errors are on the order 0.01 for $M = 2$ and below 0.005 for $M = 16$. Note that training and test accuracies can display a weak maximum in the course of learning. Therefore, for each M, we determine the number of epochs that yields the best mean training accuracy and display the corresponding test accuracy in the right panel of Fig. 3.1. The non-monotonic behavior could be cured by means of a proper regularization of GMLVQ, see (Schneider et al. 2010). Here, we resort to the above described early stopping technique for simplicity. We would like to point out that it relies only on the observed training accuracy and does not make use of test set information.

Fig. 3.1 also displays the relevance matrices and their eigenvalue spectra corresponding to the early stopping performances. In the case $M = 16$ we observe that only about $9 - 10$ eigenvalues remain significantly different from zero. Even GM-

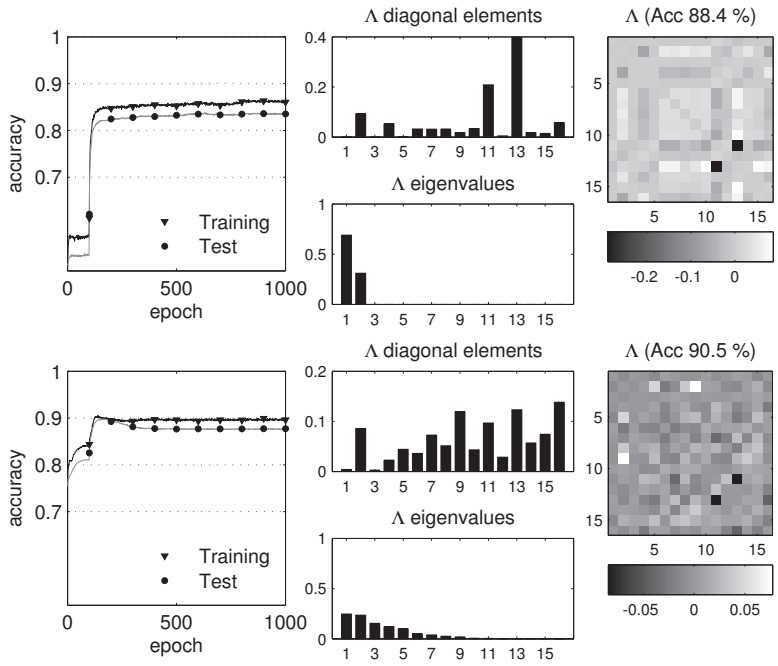


Figure 3.1: Left panels: learning curves of LiRaM LVQ with one prototype per class for $M = 2$ (top) and $M = 16$ (bottom) when applied to the UCI image segmentation data set. Right panels: diagonal elements, eigenvalues and off-diagonal elements of the matrix Λ as obtained in a single run. The diagonal elements are set to zero for the matrix plots.

LVQ with unrestricted rank results in an effective low-dimensional representation of the data. One would expect that LiRaM LVQ with large enough M already yields the same performance as the unrestricted variant. Fig. 3.2 shows that this is indeed the case. Only for small M we observe a clear dependence of the test accuracy on the rank of Ω, while all $M \geqslant 5$ display essentially the same performance. In the extreme case $M = 2$ we observe a significant drop of the generalization ability due to the serious restriction to only two non-zero eigenvalues of Λ. At the same time, the outcome of training displays a large variability: random initializations of Ω can lead to the selection of very different transformation matrices as reflected in the increased standard deviation. Many nonlinear dimension reduction methods such as Stochastic Neighbor Embedding (SNE) do not lead to a unique solution, a data set may visualized differently by the same technique in different runs. It can be argued (see e.g. (van der Maaten and Hinton 2008)) that this effect is desirable since it mir-

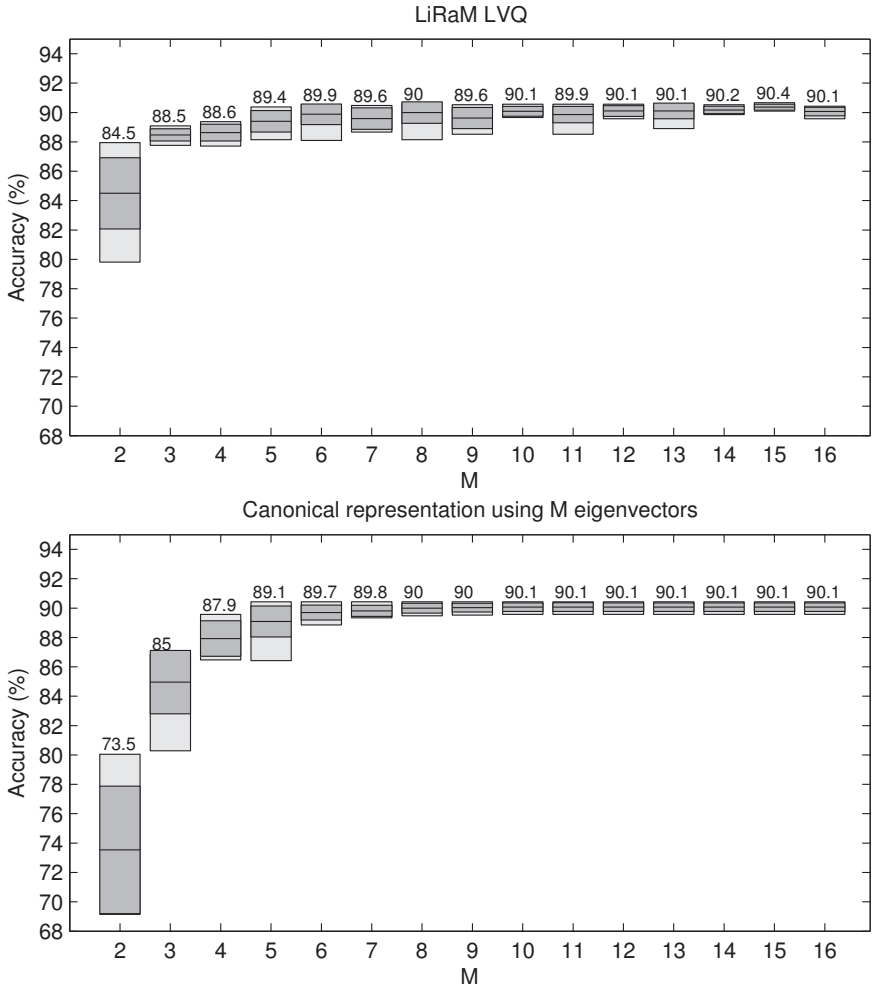


Figure 3.2: Performance of the LiRaM LVQ (upper panel) and GMLVQ with successive matrix reduction following Eq. (3.2) (lower panel) using one prototype per class as a function of M for the UCI image segmentation data set. We display the test accuracy on average over 10 random initializations, also given as a numerical value. The light shading corresponds to the interval from worst to best accuracy, the darker area marks the standard deviations.

rors different possible views of the given data and the ill-posedness of the problem of dimension reduction. Auxiliary information in the form of class labels can be useful to shape the problem in such settings and to resolve (parts of) the ambiguities

inherent in the problem. However, if the intrinsic dimension of the data is larger than the target dimension some ambiguities may not be resolved.

Additionally, we investigate the performance of the full matrix system reducing the rank after training with Eq. (3.2) using only the first M eigenvalues and eigenvectors. The lower panel of Fig. 3.2 shows the test accuracies using the $M = 16$ matrices and the canonical representation with M eigenvectors for different values of M. As observed before, keeping less than the 5 eigenvalues in the successive restricted GMLVQ (lower panel of Fig. 3.2) results in a decrease of the classification accuracy. The drop of accuracy is especially significant when eigenvectors with relatively large eigenvalues are omitted. Just using the eigenvectors of the two largest eigenvalues for example shows a mean test accuracy which is 11 % smaller than the corresponding LiRaM LVQ result for $M = 2$. Despite the computation time and memory efficiency, the limited rank version yields better preservation of the classification performance in the restricted setting than the heuristic dimension reduction after training.

3.3.2 Comparison with other methods

Here we compare the LiRaM LVQ scheme with frequently used standard procedures of comparable complexity. Note, that the complexity of LiRaM LVQ can be easily controlled by the number of prototypes. GMLVQ with only one prototype per class appears to be similar in spirit to the well known LDA (Duda et al. 2000, Friedman 1989, Bensmail and Celeux 1996). In this method, a Multivariate Normal density (MVN) is fitted to the observed data in each class, here we consider a pooled estimate of the covariance matrix. Given the density estimates, the best linear decision boundaries are constructed in order to approximate Bayes optimal classification (Duda et al. 2000). The well known Nearest Neighbor (1-NN) classifier serves as a second reference: Based on the standard Euclidean distance measure, any feature vector is simply assigned to the class of the closest labeled example (Duda et al. 2000). For the given data set, the extension to k-NN schemes displays only a weak dependence on k and results will not be presented here.

The most common strategy for dimension reduction is Principal Component Analysis (PCA). In order to compare with LiRaM LVQ, we apply PCA to the entire data set and obtain a low-dimensional representation in terms of the first M principal components. The projected training data is then used in LDA or serves as the reference set of the 1-NN classifier. In the case $M = 16$, the full data set is employed without performing a PCA.

In Fig. 3.3, the achieved test accuracies are displayed for several values of M. For large enough dimension M, the principal components capture all relevant in-

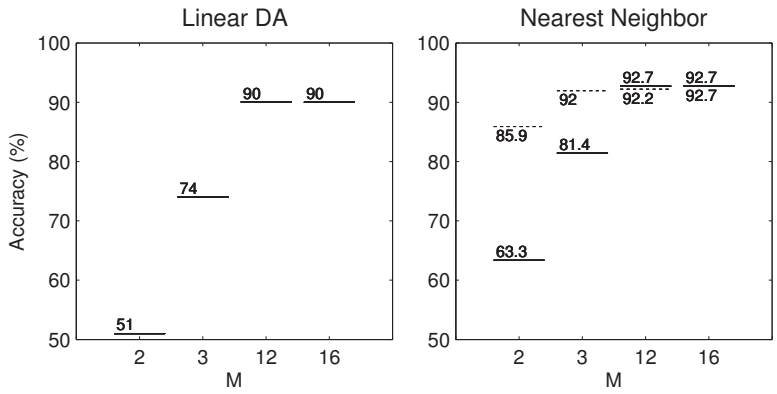


Figure 3.3: UCI image segmentation data set. Left panel: test accuracy obtained by LDA as described in the text. Right panel: test accuracies for the 1-NN classifier using the PCA-based transformation to M dimensions (solid lines). In addition, the results after transforming the data with Ω as obtained in LiRaM LVQ, the dotted lines mark the average over 10 random initialization as in Fig. 3.2.

formation and the performance of, both, LDA and 1-NN is comparable to that of the LiRaM LVQ prescription. This finding is consistent with the M-dependence discussed in the previous section.

Significant differences can be observed for small M: The dimension reduction by PCA (or any other unsupervised technique) does not take into account label information and may focus on features with large variation but little relevance for the classification. Therefore, the subsequent supervised training does not reach the quality of the LiRaM LVQ scheme even with only one prototype per class. Here, the complexity of the system is similar but the identification of a suitable low-dimensional representation is directly guided by the classification, which facilitates superior performance. This is easily demonstrated by replacing the PCA based transformation by the matrix Ω obtained in LiRaM LVQ, see Eqs. (2.13) and (3.1). Now, the simple 1-NN system performs significantly better, as displayed in the right panel of Fig. 3.3. The idea of determining a discriminative transformation directly within the k-NN classification scheme has been put forward in LMNN (Weinberger et al. 2006), there without considering dimensional reduction. A more detailed comparison of LMNN with LiRaM LVQ is given in (Bunte, Biehl, Jonkman and Petkov 2011) and in Chapter 4 of this thesis.

LiRaM LVQ with several prototypes per class and a global relevance matrix can implement piecewise linear decision boundaries, the complexity of which can exceed that of LDA or similar methods significantly. In previous applications of un-

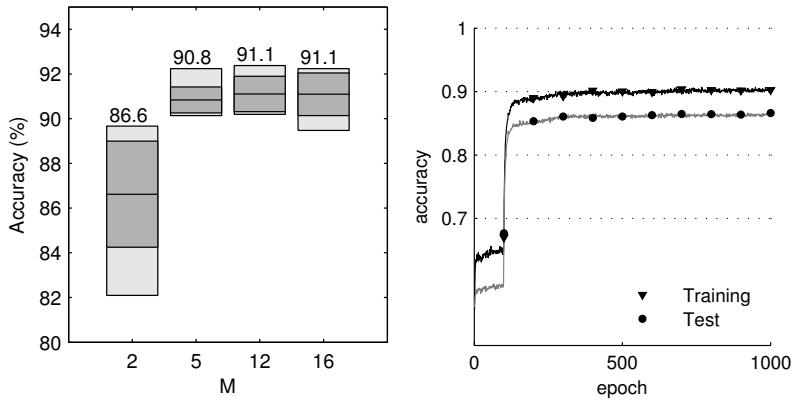
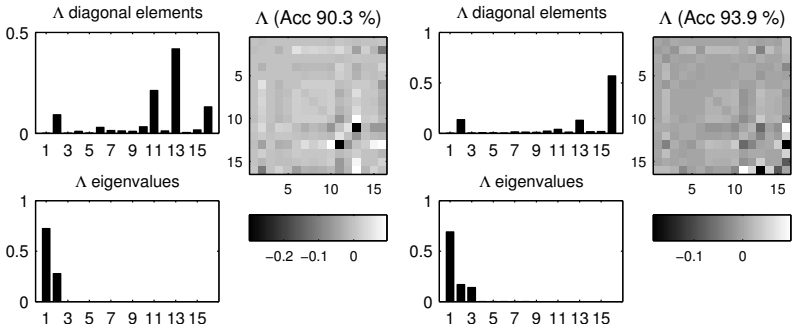


Figure 3.4: UCI segmentation. Left panel: test accuracies achieved by LiRaM LVQ with 2 prototypes per class (3 in class 5) for different values of M; other details as in Fig. 3.2. Right panel: the corresponding learning curves for $M = 2$, i.e. mean training and test accuracy vs. the number training epochs.



(a) $M = 2$ (b) $M = 16$

Figure 3.5: Diagonal elements, eigenvalues, and off-diagonal elements of an example relevance matrix in LiRaM LVQ with two prototypes per class and three in class 5. Other details as in Fig. 3.1, right panels. The diagonal elements are set to zero for the plots of the relevance matrices.

restricted GMLVQ to the UCI image segmentation data it has proven advantageous to assign 3 prototypes to class 5 (window) and 2 prototypes to all other classes. Fig. 3.4 shows that this setting improves the classification accuracies in comparison to the above studied case of a single prototype per class, cf. Fig. 3.2. As expected, the improvement is particularly pronounced for small M. In Fig. 3.5 we visualize

typical properties of the relevance matrices obtained in the extreme cases $M = 2$ and $M = 16$. Note that even the unrestricted matrix displays only three non-zero eigenvalues. The increased complexity due to the larger number of prototypes facilitates good performance in spite of a very simple implicit representation of the data. The use of more eigendirections could be enforced by means of a matrix regularization scheme suggested in (Schneider et al. 2010). We will address this issue in forthcoming studies.

3.4 Visualization of classification schemes

The LiRaM LVQ prescription with $M = 2$ or $M = 3$ can be readily employed as a tool for the visualization of labeled data sets. In contrast to many standard methods, the tasks of identifying an appropriate subspace and implementing the actual classification is addressed in a single training phase. Supervised dimension reduction has drawn some attention recently, some of the methods have been mentioned in the Introduction. We explain two of these methods in the next section in more detail and will compare example visualizations of different data sets thereafter.

3.4.1 Local Fisher Discriminant Analysis

A supervised linear dimension reduction technique named LFDA (Sugiyama and Roweis 2007) was recently introduced as a combination of the well known Fisher Discriminant Analysis (FDA) (Fisher 1936) and the unsupervised Locality-Preserving Projection (LPP) (He and Niyogi 2003). FDA works particularly well, when each class can be modeled as an unimodal Gaussian. It is based on the within-class and between-class scatter matrix and finds a transformation matrix T, such that the between-class scatter is maximized, while the within-class scatter is minimized. This optimization problem can be solved by means of a generalized eigenvalue problem (Fukunaga 1990). The between-class scatter matrix has a rank limited to the number of classes minus one ($C - 1$). This implies that FDA can find at most $C - 1$ meaningful features, which constitutes a serious restriction in practice. LPP on the other hand is an unsupervised dimension reduction technique based on pairwise affinities $A_{i,j} \in [0, 1]$ between data points $\boldsymbol{x}^i$ and $\boldsymbol{x}^j$. The aim is to find a transformation matrix T such that local neighborhoods are preserved in the embedding space.

The LFDA efficiently combines the ideas of both methods and facilitates the dimension reduction of multi-modal labeled data by maximizing the between-class separability, while preserving the local structure within classes. The local within-class and local between-class scatter matrices $S^{(w)}$ and $S^{(b)}$ are defined using pair-

wise affinities of the data:

$$S^{(w)} = \frac{1}{2} \sum_{i,j=1}^{n} W_{i,j}^{(w)} (\boldsymbol{x}^i - \boldsymbol{x}^j)(\boldsymbol{x}^i - \boldsymbol{x}^j)^\top \tag{3.11}$$

$$S^{(b)} = \frac{1}{2} \sum_{i,j=1}^{n} W_{i,j}^{(b)} (\boldsymbol{x}^i - \boldsymbol{x}^j)(\boldsymbol{x}^i - \boldsymbol{x}^j)^\top \ , \tag{3.12}$$

where n denotes the number of samples and

$$W_{i,j}^{(w)} = \begin{cases} A_{i,j}/n_l & \text{if } y^i = y^j = l \\ 0 & \text{if } y^i \neq y^j \end{cases} \tag{3.13}$$

$$W_{i,j}^{(b)} = \begin{cases} A_{i,j}(\frac{1}{n} - \frac{1}{n_l}) & \text{if } y^i = y^j = l \\ 1/n & \text{if } y^i \neq y^j \end{cases} . \tag{3.14}$$

The value n_l denotes the number of samples from class l. Therefore, LFDA aims at finding a transformation matrix T, such that nearby data pairs of the same class are also close in the embedding and data points of different classes are separated from each other. Similar to FDA also LFDA projection can be computed analytically by solving a generalized eigenvalue problem:

$$T = \underset{T \in \mathbb{R}^{N \times M}}{\operatorname{argmax}} \left[tr \left((T^\top S^{(w)} T)^{-1} T^\top S^{(b)} T \right) \right]. \tag{3.15}$$

In contrast to FDA the LFDA does not have the same rank limitation. Therefore a dimension reduction to arbitrary dimensions is possible. However, the embedding crucially depends on the computation of the pairwise affinities. In (Sugiyama and Roweis 2007) four definitions of the affinity matrix are given. In the following experiments we use the "local scaling" method, which is also used in the provided implementation[1]. Here the density of the data is taken into account in a heuristic manner: a local scaling based on the k-th nearest neighbor is included. In the experiments we tried different values of k to find good visualizations.

3.4.2 Neighborhood Component Analysis

Recently, a supervised dimension reduction method called NCA has been introduced (Goldberger et al. 2004). It aims in the maximization of the expected number of correctly classified samples by a stochastic variant of the nearest neighbor clas-

[1] MATLAB implementation LFDA: http://sugiyama-www.cs.titech.ac.jp/~sugi/software/LFDA/

sifier. Therefore, NCA seeks a transformation matrix T_{NCA} such that the between-class separability is maximized:

$$T_{\mathrm{NCA}} = \underset{T \in \mathbb{R}^{N \times M}}{\operatorname{argmax}} \left(\sum_{i=1}^{n} \sum_{y^j = y^i} p_{i,j}^{\mathrm{NCA}} \left(T T^{\top} \right) \right) \tag{3.16}$$

where

$$p_{i,j}^{\mathrm{NCA}}(U) = \begin{cases} \frac{\exp\{-(\boldsymbol{x}^i - \boldsymbol{x}^j)^{\top} U (\boldsymbol{x}^i - \boldsymbol{x}^j)\}}{\sum_{k \neq i} \exp\{-(\boldsymbol{x}^i - \boldsymbol{x}^k)^{\top} U (\boldsymbol{x}^i - \boldsymbol{x}^k)\}} & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases} .$$

Thus, similar to LFDA, nearby data pairs from the same class should be close in the embedding space. This ensures that also multi-modal structure of the data can be preserved. However, the optimization problem is non-convex and there is no guarantee that the global optimum can be obtained. The optimization was proposed as a gradient ascent method and we use the provided implementation [2] for the experiments. Note, that NCA needs to compute the pairwise dissimilarities between samples of the same class in every step. Although LiRaM LVQ also follows a gradient procedure it computes only the dissimilarities with respect to the prototypes in every step. Since the number of prototypes per class is usually much smaller than the number of samples, the computational costs per gradient step are significantly lower than for NCA. In the implementation a Polack-Ribiere flavor of conjugate gradients is used to compute search directions, and a line search using quadratic and cubic polynomial approximations. There is mainly one parameter to change: l the length of the run. It corresponds to the maximum number of line searches.

3.4.3 The segmentation data set

The above discussed UCI segmentation data may serve as a first illustrative example. From the 10 independent runs performed with $M = 2$ to obtain the results displayed in Fig. 3.2 (single prototype per class) and Fig. 3.4 (several prototypes per class), we have selected the runs that achieved the best training accuracy in order to achieve the most discriminative visualization. As mentioned above, the actual outcome can depend on the random initialization of the LiRaM LVQ system, see Figs. 3.2 and 3.4 for the range of observed accuracies. With a single prototype per class, a maximum classification accuracy of 88.4% on the entire data set is achieved. The use of 2 prototypes per class (3 in class 5) yields a best accuracy of 90.4% on the

[2] MATLAB implementation for NCA: http://www.ics.uci.edu/~fowlkes/software/nca/

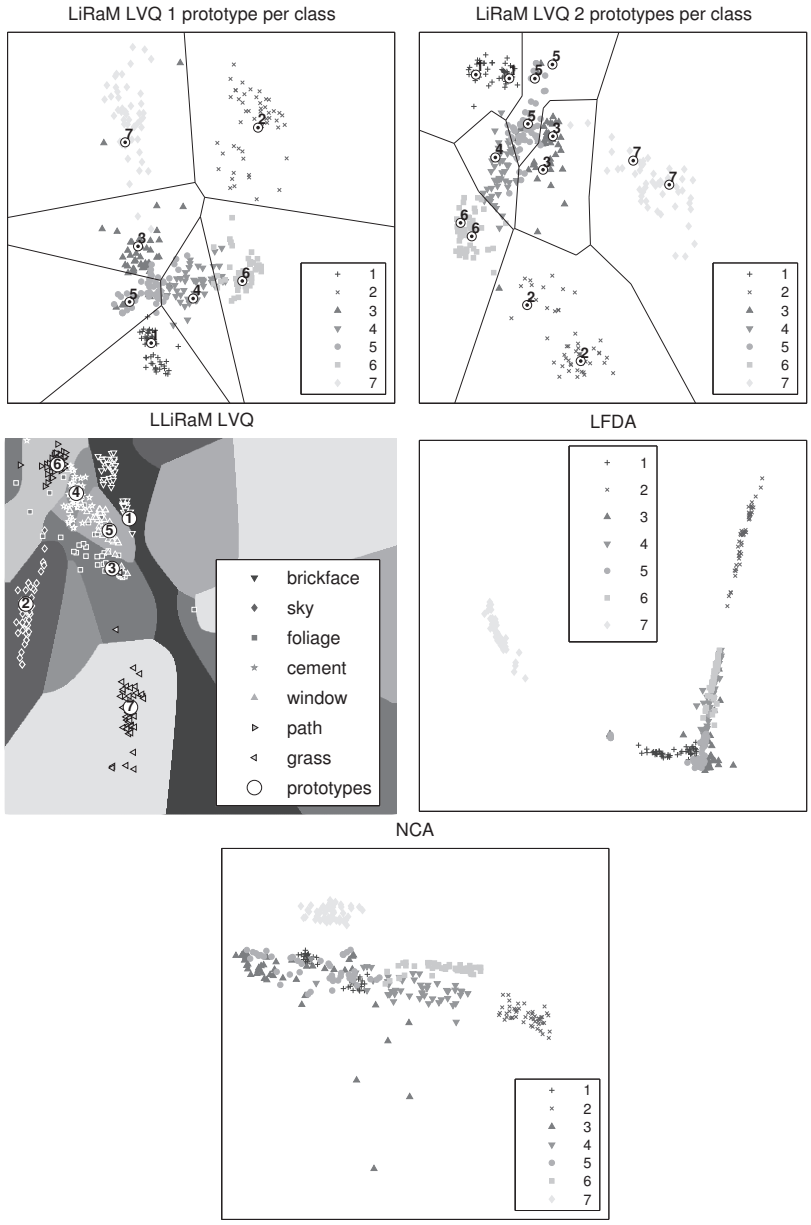


Figure 3.6: Visualizations of the UCI segmentation data set aquired by the different methods. For the sake of clarity we display only 50 examples per class. Detailed explanation can be found in the text.

entire set. The use of several prototypes with LLiRaM LVQ enhances the accuracy by realizing more complex piecewise linear decision boundaries.

Furthermore we trained the LLiRaM LVQ under the same conditions ten times on the training set of the segmentation data and used the resulting transformations and prototypes to visualize the data. The run showing the best performances is shown in Fig. 3.6 with the quality given in Table 3.1. The mean accuracy over all runs on the training data is 85% with a standard deviation (STD) of 0.04 with one prototype per class and class-wise dissimilarities Ψ^c. LLiRaM LVQ implements non-linear decision boundaries, which shows already good accuracies using one prototype per class. With this particular data set using more prototypes does not improve the classification significantly.

Additionally, we employ the implementation of LFDA and NCA from the original authors with default parameters and tried a range of k and $l \in [1, 30]$. We observed, that both methods crucially depend on the parameter used. The accuracy on the training set measured by an 1-NN classification on the embedding aquired by LFDA, for example, ranges from the best accuracy 83.7% with $k = 2$ and the worst accuracy 66.6% with $k = 25$. For NCA the worst accuracy of 56.2% is observed with $l = 1$ and with $l \geqslant 16$ the training accuracy reaches 90%. The number of protoytpes and the initialization in the LiRaM LVQ setting is less crucial with respect to the classification accuracy.

Fig. 3.6 displays the best visualizations of the segmentation data set aquired by the different techniques explained above. This multi-class problem allows for very good classification performance already in two dimensions. The localized variant of LiRaM LVQ can realize more complicated non-linear decision boundaries than the global version. However, overfitting effects become possible: For one prototype per class we observe an improvement although empty cells appear in the tessellation. With two prototypes per class no further improvement is observed. In all visualizations the classes "sky" and "grass" can be separated quite well. For the other classes the visualizations differ in arrangement and shape of the clusters. The LiRaM LVQ visualizations show equal or superior quality compared to the other methods. An overview of the visualization quality of the different methods on the data sets can be found in Table 3.1. The classification accuracy in the original space is usually larger, than the accuracy in the low-dimension space after transformation. However, the numbers show, that in most cases the supervised dimension reduction was able to preserve high accuracies even in the reduced spaces. We would like to point out once more, that the computational effort for NCA is much larger than for the LiRaM LVQ variants. NCA computes all pairwise distances, while the LVQ approaches are based on a small number of prototypes. In particular for large data sets the computational effort may be reduced significantly compared to NCA.

Table 3.1: Classification and 1-NN accuracies (acc. in %) on the visualizations of the data sets. The quantity P denotes the number prototypes.

method / data set		acc. training	acc. test
Segmentation data			
LiRaM LVQ 7P	(classification accuracy)	92.9	88.0
LiRaM LVQ 7P	(1-NN acc. on embedding)	85.7	87.0
LiRaM LVQ 14P	(classification accuracy)	91.9	90.3
LiRaM LVQ 14P	(1-NN acc. on embedding)	88.6	87.5
LLiRaM LVQ	(classification accuracy)	89.0	85.7
LLiRaM LVQ	(1-NN acc. on embedding)	88.6	87.4
LFDA	(1-NN acc. on embedding)	83.7	85.8
NCA	(1-NN acc. on embedding)	90.0	87.1
Colorado data 2D			
LiRaM LVQ	(classification accuracy)	83.0	80.0
LiRaM LVQ	(1-NN acc. on embedding)	79.6	84.6
LLiRaM LVQ	(classification accuracy)	78.7	73.8
LLiRaM LVQ	(1-NN acc. on embedding)	79.9	83.7
LFDA	(1-NN acc. on embedding)	50.4	61.1
NCA	(1-NN acc. on embedding)	81.5	89.7
Colorado data 3D			
LiRaM LVQ	(classification accuracy)	88.9	86.3
LiRaM LVQ	(1-NN acc. on embedding)	93.3	96.4
LLiRaM LVQ	(classification accuracy)	87.7	85.8
LLiRaM LVQ	(1-NN acc. on embedding)	92.8	96.1
LFDA	(1-NN acc. on embedding)	89.6	93.8
NCA	(1-NN acc. on embedding)	92.6	95.5

3.4.4 High-dimensional Gene Expression Data

Discriminative visualization can be particularly useful in the context of medical data. Here we apply the LiRaM LVQ algorithm to two gene expression data sets which were recently analyzed in (Faith et al. 2006). The first set concerns *small round blue cell childhood tumors,* and we refer to it as SRBCT (Faith et al. 2006). It comprises cDNA microarray expression levels of 50 pre-selected genes in 83 different samples (Khan et al. 2001). The target classification assigns every sample to one of 4 tumor types. We will refer to the second data set as NCI. It contains gene expression data from 60 cell lines from the National Cancer Institute anticancer drug screen (Scherf et al. 2000). Again 50 genes have been pre-selected and samples are to

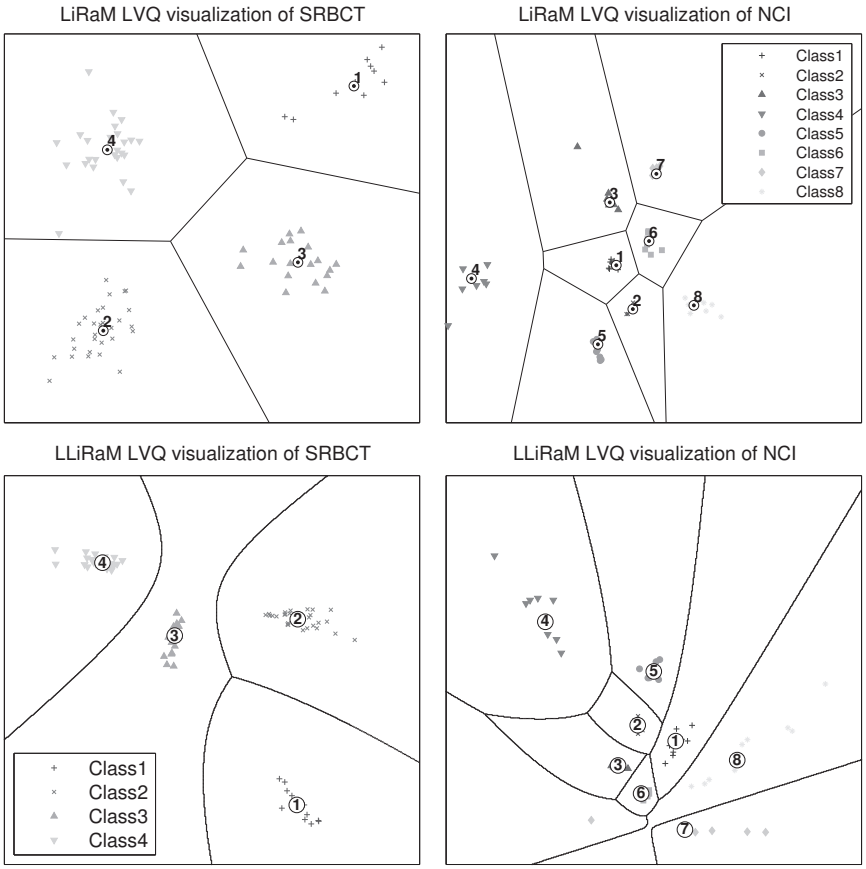


Figure 3.7: Two-dimensional, visualizations of the SRBCT data set (left column) and the NCI data (right column) obtained by the different variants of LiRaM LVQ explained in the text.

be assigned to one of 8 different types of tissue. For details of the data sets we refer to (Faith et al. 2006) and references therein. The authors present a method termed Targeted Projection Pursuit (TPP) and compare it with several existing techniques, including MDS (Ewing and Cherry 2001), VizStruct (Zhang et al. 2004), a dendrogram based method (Eisen et al. 1998), and Projection Pursuit (Lee et al. 2005). TPP is demonstrated to outperform most of these methods or to achieve at least comparable performance on the above data sets. The employed data sets as well as source codes of TPP implementations are publicly available (Faith et al. 2006). First, we apply LiRaM LVQ with one prototype per class to the SRBCT data set. Results pre-

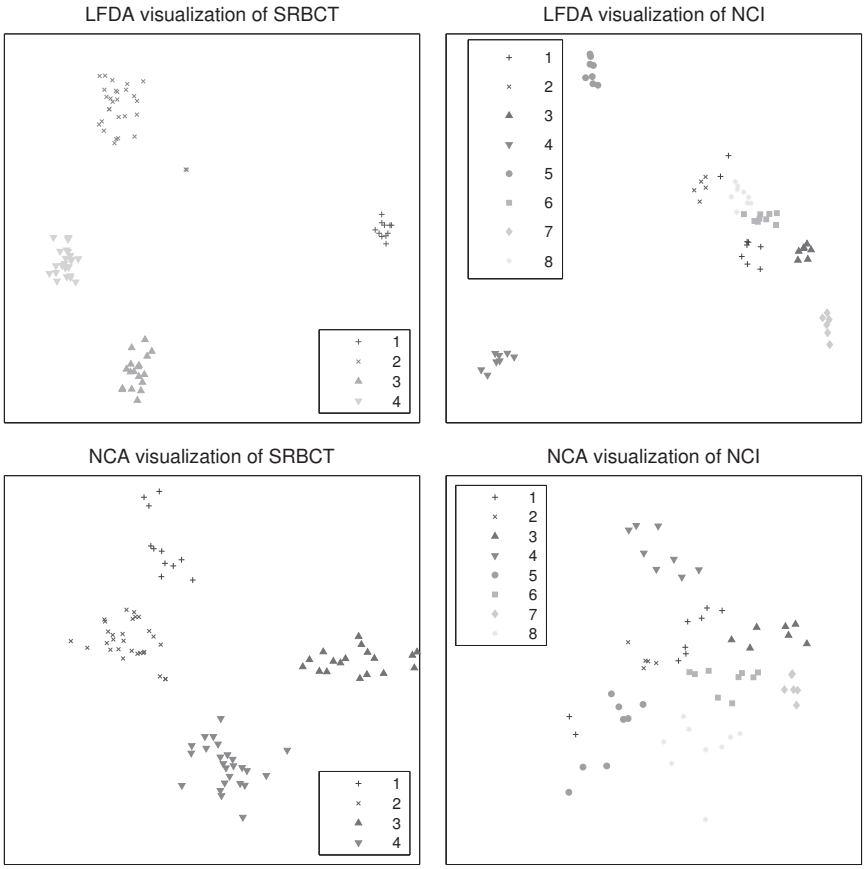


Figure 3.8: Two-dimensional, visualizations of the SRBCT data set (left column) and the NCI data (right column) obtained by LFDA and NCA. Detailed explanation can be found in the text.

sented here are obtained after 1000 epochs with respect to the entire data set of 83 samples. We observe almost no variability with respect to random initializations of the system. A typical outcome is displayed in Fig. 3.7 (top row left panel) the obtained $2D$ visualization perfectly separates the four classes. Error free visualizations were also obtained by Faith et *al.*, see (Faith et al. 2006) for comparison.

The analogous application of LiRaM LVQ to the NCI 8-class-problem shows slightly larger variability of results. In 10 runs with different random initialization we obtain after 1000 epochs accuracies in the range from 95.1%-100%, with an average of 97.7%. Fig. 3.7 (upper row, right panel) displays a perfectly separating

visualization. For the sake of completeness we show the error-free example results of the LLiRaM LVQ with one prototype per class in Figure 3.7 (bottom row). The algorithm was trained with the same parameters as the global version on both, the whole SRBCT and NCI, data set. Again the four-class problem SRBCT can be separated in every Run with random initialization, whereas the training on the NCI data set shows some variation in classification accuracy. In mean we achieved on the NCI data an accuracy of 94.6% with a standard deviation of 0.02 over the 10 random initializations.

The visualization of these data sets achieved by LFDA and NCA are shown in Fig. 3.8. LFDA was performed on the SRBCT data set with $k \in [1, 10]$, all yielding error free visualizations. On the NCI data set the accuracy varied from 91.8% achieved with $k = 4$ to the best accuracy of 96.7% using $k = 1$. For the training of NCA on the SRBCT data set with l varying from one to 10 we observed error free visualizations for $l \geqslant 3$ and the worst accuracy of 80.7% for $l = 1$. On the NCI data set an error free visualization is found for $l \geqslant 10$ and the worst performance was 59% observed with $l = 1$. In (Faith et al. 2006), error free visualizations of the NCI data are obtained by means of TPP in combination with PCA, Projection Pursuit and subsequent LDA or k-NN classification. For a visual inspection of the achieved separation we refer to Figs. 9 and 11 in (Faith et al. 2006), which display either slightly overlapping classes or only very small gaps between some of them. Other methods considered in (Faith et al. 2006) yield less favorable results on this data set. Most of all, we would like to point out that our method appears very simple and intuitive compared to many other suggested approaches. However, it yields comparable or even superior results at comparably low computational costs.

3.4.5 Satellite Remote Sensing data

Here we apply the algorithm to a large real world data set: a multi-spectral satellite image of the Colorado area, focusing of visualizing the class structure. Remote sensing spectral images consist of an array of multi-dimensional vectors (spectra) assigned to particular spatial regions (pixels) reflecting the response of a spectral sensor at various wavelengths. A spectrum is a characteristic pattern that provides a clue to the surface material within the respective area. The use of these data includes areas such as mineral exploration, land use, forestry; and many other activities of economic significance.

We consider a data set that corresponds to an image taken close to Colorado Springs using satellites of the LANDSAT-TM type. The size of the image is 1907×1784 pixels, each of which corresponds to an area of $900m^2$ on the ground. The spectrum is represented by a 6-dimensional feature vector. The aim of the classification

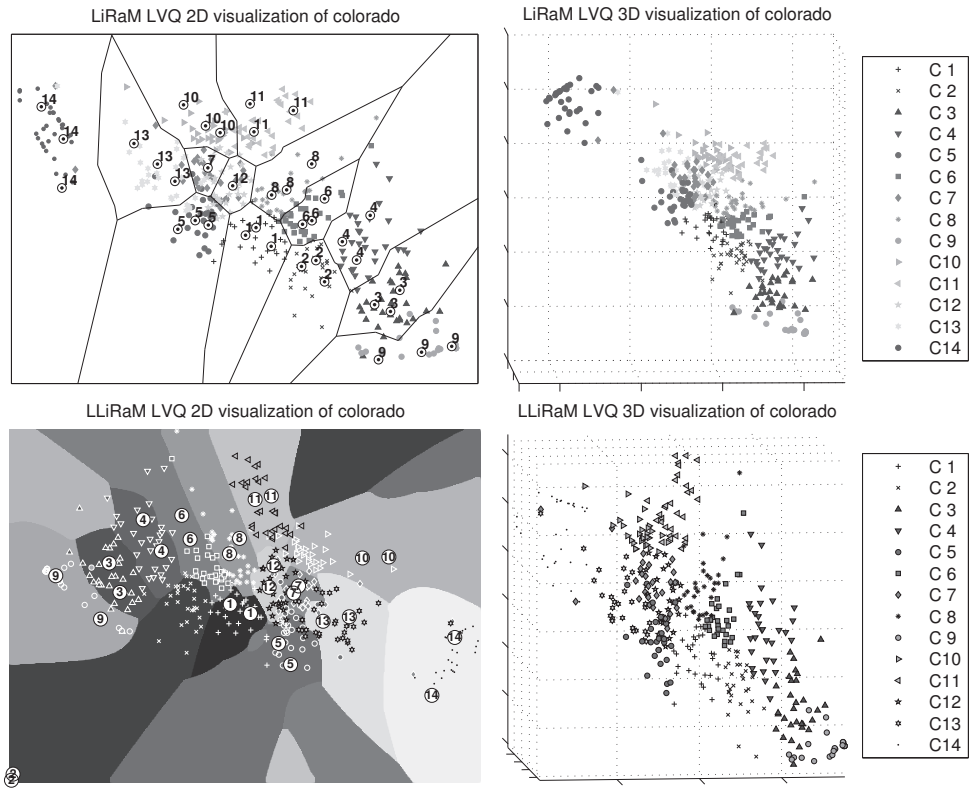


Figure 3.9: Visualizations of a small subset of the Colorado data set aquired by the different methods.

is to assign each pixel to one of 14 classes, corresponding to specific surface covers such as different types of forests, alpine vegetation, water, etc., see (Hammer and Villmann 2002, Villmann et al. 2003) for a detailed description and Table 3.2 for the list of classes. A labeling of the entire image was provided by experts and serves as the target classification. For further details of the data set we refer the reader to (Hammer and Villmann 2002, Villmann et al. 2003) where the authors apply scaled Euclidean distance in combination with a Growing Self-Organized Map (GSOM). Test accuracies in the range of 90% have been achieved depending on the specific method in use.

For the following, we selected 2000 examples per class randomly, used as a training set. We also give the accuracies evaluated with respect to the whole data set of 3,402,088 data points. We have performed 10 runs of LiRaM LVQ with $M = 2, 3$ and

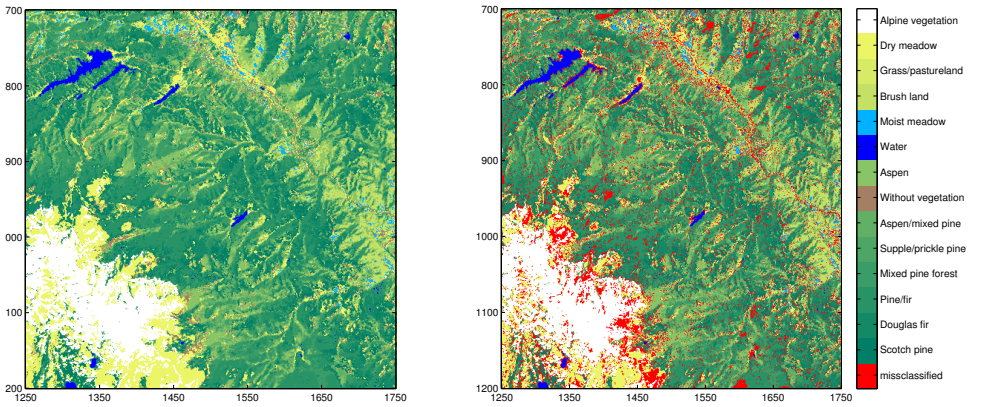


Figure 3.10: The labels of a section of the Colorado satellite image (left panel) and the classification result obtained by the best run of LLiRaM LVQ in the 3D case (right panel). Detailed information about the class-wise accuracies can be found in the confusion matrix Tab. 3.3.

Table 3.2: Short description of the different classes of the satellite image and the number of pixels in each class.

class	ground cover type	# pixels
1	Scotch pine	581424
2	Douglas fir	355145
3	Pine / fir	181036
4	Mixed pine forest	272282
5	Supple/prickle pine	144334
6	Aspen/mixed pine forest	208152
7	Without vegetation	170196
8	Aspen	277778
9	Water	16667
10	Moist meadow	97502
11	Bush land	127464
12	Grass/pastureland	267495
13	Dry meadow	675048
14	Alpine vegetation	27556
0	not classified	9

three prototypes per class. After 1500 training epochs we observe only very little variation due to the random initialization of the system. The range of training accuracies is 79.8%-83% for $M = 2$ and 87.5%-88.9% for $M = 3$, respectively. The classifiers with the best training set performance achieve accuracies on the whole set of 80.1% ($M = 2$) and 86.3% ($M = 3$), see Table 3.1. In spite of the low-dimensional representation and the relatively small numbers of prototypes we achieve very good accuracies. This is consistent with the analysis in (Villmann et al. 2003) which suggests that good classification performance requires at least a two- or three-dimensional representations of the data.

Here, we are mainly interested in the discriminative visualization of the data set. Fig. 3.9 shows the data globally projected into two and three dimensions,

Table 3.3: Confusion matrix of the 3D LLiRaM LVQ on the Colorado data set.

	actual class															
C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	0	$\sum$
1	460594	612	104	5	2376	458	49	883	4	0	0	1498	0	139	0	466722
2	13642	331530	590	11146	0	841	9	79	8	0	0	0	0	0	0	357845
3	0	9379	155775	17306	0	0	1	0	757	0	0	0	0	0	0	183218
4	0	3742	704	231063	0	596	1	7	90	0	0	0	0	0	0	236203
5	14776	0	11	0	122956	0	7793	0	1	0	0	2989	25239	70	0	173835
6	22880	8618	102	12235	5	203917	7	7980	28	0	0	0	0	0	0	255772
7	521	0	3	3	7337	0	111692	360	3	66	554	23873	31728	0	0	176140
8	18380	0	60	14	41	2340	11	256243	8	1	1597	10277	0	0	1	288973
9	14	1210	23613	479	143	0	46	0	15761	0	0	0	0	116	0	41382
10	3	0	5	7	38	0	12842	0	1	86795	7970	7894	7352	0	0	122907
11	0	0	18	11	0	0	285	11660	0	6508	117212	4352	0	0	0	140046
12	48564	54	38	5	8716	0	24687	566	3	2279	130	216576	10522	0	0	312140
13	2045	0	13	8	2611	0	4063	0	3	1853	0	36	582457	148	1	593238
14	5	0	0	0	111	0	8710	0	0	0	1	0	17750	27083	7	53667
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\sum$	581424	355145	181036	272282	144334	208152	170196	277778	16667	97502	127464	267495	675048	27556	9	3402088
class-wise accuracy of the estimation in %																
	79.22	93.35	86.05	84.86	85.19	97.97	65.63	92.25	94.56	89.02	91.96	80.96	86.28	98.28	0	

respectively. We also trained the Localized LiRaM LVQ on 2000 random samples from each class with slightly different parameters: 300 epochs, learning rates beginning with τ^{start}=0.001 and $\Delta\tau = 0.0001$ for the prototypes, the matrix Ω and the class-wise matrices Ψ^c respectively. We trained the system with two and three prototypes per class. The average accuracy on the training data is 75% with STD 0.03 in the two-dimensional case with 28 prototypes. In three dimensions with three prototypes per class we obtain a mean accuracy of 85.2% and STD 0.02. These results correspond to the findings in (Hammer and Villmann 2002) where GRLVQ was applied to the data set: When pruning to three dimensions a classification performance of ca. 84% can be achieved, while dropping further dimensions decreases the accuracy significantly. The visualizations resulting from the best run in two and three dimensions are shown in Fig. 3.9 (bottom row). Furthermore, the confusion matrix for the three-dimensional case containing information about the class-wise accuracies and misclassification can be found in Table 3.3. We also provide the original labeling of the satellite image and the estimated Labels with misclassification. The corresponding graphics can be found in Fig. 3.10. The projections facilitate a detailed interpretation and analysis of the data set. We will present and exploit the obtained insights in a forthcoming study.

We demonstrate the advantages of LiRaM LVQ and its localized variant over LFDA and NCA: Fig. 3.11 shows the best visualizations we could achieve with this methods. We varied the value k and l in the interval $[1, 10]$ and for LFDA we achieved the best 1-NN error measures on the visualizations with $k = 6$ and $k = 9$ for 2D and 3D respectively. While certain classes (e.g. 14, alpine vegetation) seem to separate well, the overall discriminativity is limited. Only 50.4% accuracy can be

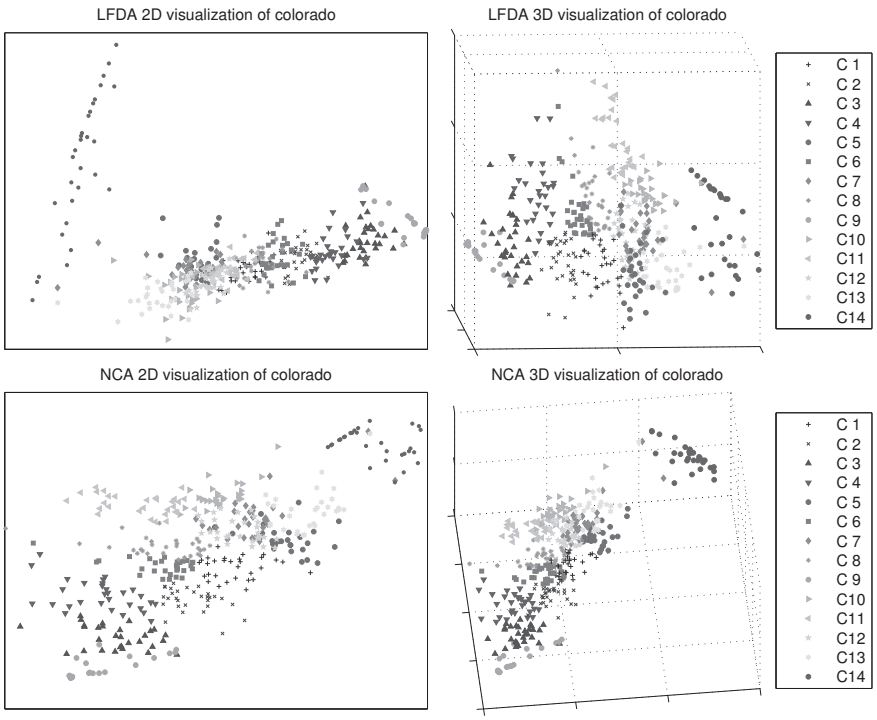


Figure 3.11: Visualizations of a small subset of the Colorado data set aquired by the different methods.

achieved using a 1-NN classifier on the training data in the two-dimensional case and 89.6% in the three-dimensional case. For this particular data set the value of the parameter k has no significant influence on the quality of the LFDA-embedding of the training data. The computation of the 1-NN error on over three million data points of the test set was not practicable. Therefore we draw 100 000 points randomly from the test set and this reduced set serves as approximation of the test-error. With the best LFDA we observed 61.3% and 93.75% 1-NN classification accuracy on the reduced test set for two and three dimensions, respectively. Table 3.1 shows the detailed comparison. The use of NCA turned out in-practicable due to excessive memory use. Therefore, we reduced the training set to 900 samples per class. We tried different values for the parameter l ranging from one to ten. The best results are shown in Fig. 3.11 (bottom row) for $k = 3$ and $k = 2$ in the 2D and 3D visualization respectively. On this data set the best NCA parametrization showed comparable or even better results than the LVQ approach. Nevertheless, some pa-

tience was necessary to get these results due to the computational complexity and the variation with respect to the parameter was huge. In the two-dimensional case the 1-NN accuracy ranged between 56.43% and 81.49% on the training set and in the 3-dim. case accuracies between 67.29% and 92.56% were observed. The other methods showed to be faster and more robust with respect to the parametrization.

3.5 Summary and outlook

In this Chaper we present the LiRaM LVQ algorithm together with a localized variant, as a modification of GMLVQ (Schneider et al. 2009a). It employs rectangular projection matrices to represent N-dim. feature vectors in an M-dim. space internally. This makes it possible to limit the rank of the relevance matrices used in GMLVQ which parameterize an adaptive distance measure. Obvious aims are to incorporate prior knowledge of the intrinsic dimension or to reduce the number of free parameters while maintaining good classification performance. In particular for high-dimensional data sets this can reduce the computational effort significantly. First we illustrate the approach in terms of a multi-class benchmark data set and compare with other methods of similar complexity. We demonstrate that LiRaM LVQ is an efficient method for determining discriminative, low-dimensional representations of labeled data and facilitates good generalization behavior. In LiRaM LVQ, the search for the appropriate subspace is guided directly by the classification performance in a single supervised training phase. This is in contrast to classical combinations of unsupervised dimension reduction and subsequent supervised learning.

A particular attractive application of the concept concerns the visualization of labeled data sets. Setting $M = 2$ or 3 in LiRaM LVQ provides us with a discriminative visualization of the original data set. The algorithm results in linear or piecewise linear decision boundaries dependent on the number of prototypes and classes. With the localized variant LLiRaM LVQ it is possible to visualize even more complicated non-linear decision boundaries. We demonstrate the usefulness of this concept in the context of several real world multi-class problems. Furthermore we compare the visualizations to some recent state-of-the-art supervised dimension reduction techniques, namely LFDA and NCA. The LFDA approach provides an analytical solution, but also depends on the computation of pairwise dissimilarities within samples of the same class. The results may differ a lot depending on the number k of neighbors used. For less complex data sets, like the four class SRCBT cancer data set error free visualizations are possible. On other data sets LFDA showed worse results compared to the other methods. NCA showed good results in most cases.

Its performance is also dependent on random initialization and the number of line searches l. NCA is based on the computation of pairwise dissimilarities which is expensive for large data sets. The LiRaM LVQ approach displays in all cases comparable or superior results on the investigated data sets. The computational effort depends on the target dimension, the number of prototypes and the number of samples for training. Unlike other methods, which require all pairwise dissimilarities, LiRaM LVQ computes distances of samples with respect to only a few prototypes. The observed influence of the number of protoytpes on the performance is weak compared to the dependence on the neighborhood parameter in other methods.

The use of local or class-wise transformation matrices in LLiRaM LVQ allows for more complex decision boundaries. The decision boundary in the low-dimensional space is based on local matrices attached to the prototypes. Note, that the dimension reduction itself is done in terms of a global linear projection. The concept of using local dissimilarities in combination with non-linear dimension reduction and visualization was recently discussed in (Bunte, Hammer, Wismüller and Biehl 2010). In this paper we have not emphasized one particularly attractive feature of relevance learning: The resulting transformation and relevance matrices can be readily interpreted and carry important information about the structure of the data. For instance, in the visualization of gene expression data, Sec. 3.4.4, we note that several features (intensities) essentially do not contribute to the highly discriminative linear combinations defined by Ω. This type of information provides valid insights to the application expert and should be exploited systematically.

In forthcoming projects we will also investigate several extensions of the method. So far, we only limit the maximum rank of relevance matrices by choice of the parameter M, the effective dimension of the transformation can become even smaller. In applications, including visualization, it can be desirable to fix the rank and to make the system exhaust the bound. This could be done in terms of an efficient regularization method which we developed recently (Schneider et al. 2010). Most importantly, we plan to apply the LiRaM LVQ approach in various application domains, including the ones discussed above. An example application in the context of Content Based Image Retrieval (CBIR) is discussed in (Bunte, Biehl, Jonkman and Petkov 2011) and Chapter 4 of this thesis.

3.A Derivatives of GMLVQ and LiRaM LVQ

Here we show the derivatives of the GMLVQ costfunction E_{GMLVQ} for one presented training example $\boldsymbol{x}^i$, see Eq. (2.14), with respect to the prototypes $\boldsymbol{w}^L$ with $L \in \{J, K\}$ and the transformation matrix $\Omega \in \mathbb{R}^{M \times N}$. The derivative with respect to the prototypes can be formulated like following:

$$d_L^\Lambda = \sum_r^N \sum_m^M \sum_n^N (x_r^i - w_r^L)\Omega_{mr}\Omega_{mn}(x_n^i - w_n^L) \tag{3.17}$$

$$\frac{\partial E_{\mathrm{GMLVQ}}}{\partial \boldsymbol{w}^L} = \frac{\Phi(\mu^i)}{\partial \mu^i} \cdot \frac{\partial \mu^i}{\partial d_L^\Lambda} \cdot \frac{\partial d_L^\Lambda}{\partial \boldsymbol{w}^L} \tag{3.18}$$

$$\frac{\partial \mu^i}{\partial d_J^\Lambda} = \gamma^J = \frac{(d_J^\Lambda + d_K^\Lambda) - (d_J^\Lambda - d_K^\Lambda)}{(d_J^\Lambda + d_K^\Lambda)^2} = \frac{2d_K^\Lambda}{(d_J^\Lambda + d_K^\Lambda)^2} \tag{3.19}$$

$$\frac{\partial \mu^i}{\partial d_K^\Lambda} = \gamma^K = \frac{-(d_J^\Lambda + d_K^\Lambda) - (d_J^\Lambda - d_K^\Lambda)}{(d_J^\Lambda + d_K^\Lambda)^2} = \frac{-2d_J^\Lambda}{(d_J^\Lambda + d_K^\Lambda)^2} \tag{3.20}$$

$$\frac{\partial d_L^\Lambda}{\partial w_r^L} = -2 \cdot \sum_n^N \sum_m^M \Omega_{mr}\Omega_{mn}(x_n^i - w_n^L) = -2\left[\Omega^\top \Omega\right]_r (\boldsymbol{x}^i - \boldsymbol{w}^L) \tag{3.21}$$

$$\frac{\partial d_L^\Lambda}{\partial \boldsymbol{w}^L} = -2 \cdot \Omega^\top \Omega(\boldsymbol{x}^i - \boldsymbol{w}^L) \ . \tag{3.22}$$

The corresponding matrix update reads:

$$\frac{\partial E_{\mathrm{GMLVQ}}}{\partial \Omega_{mn}} = \frac{\Phi(\mu^i)}{\partial \mu^i} \cdot \frac{\partial \mu^i}{\partial \Omega_{mn}} \tag{3.23}$$

$$\begin{aligned}
\frac{\partial \mu^i}{\partial \Omega_{mn}} &= \frac{\left(\frac{\partial d_J^\Lambda}{\partial \Omega_{mn}} - \frac{\partial d_K^\Lambda}{\partial \Omega_{mn}}\right)(d_J^\Lambda + d_K^\Lambda) - (d_J^\Lambda - d_K^\Lambda)\left(\frac{\partial d_J^\Lambda}{\partial \Omega_{mn}} + \frac{\partial d_K^\Lambda}{\partial \Omega_{mn}}\right)}{(d_J^\Lambda + d_K^\Lambda)^2} \\
&= \frac{2d_K^\Lambda}{(d_J^\Lambda + d_K^\Lambda)^2} \cdot \frac{\partial d_J^\Lambda}{\partial \Omega_{mn}} + \frac{-2d_J^\Lambda}{(d_J^\Lambda + d_K^\Lambda)^2} \cdot \frac{\partial d_K^\Lambda}{\partial \Omega_{mn}} \\
&= \gamma^J \frac{\partial d_J^\Lambda}{\partial \Omega_{mn}} + \gamma^K \frac{\partial d_K^\Lambda}{\partial \Omega_{mn}}
\end{aligned} \tag{3.24}$$

$$\begin{aligned}
\frac{\partial d_L^\Lambda}{\partial \Omega_{mn}} &= 2\sum_r^N (x_n^i - w_n^L)\Omega_{mr}(x_r^i - w_r^L) \\
&= 2\left[\Omega(\boldsymbol{x}^i - \boldsymbol{w}^L)\right]_m \cdot (x_n^i - w_n^L) \ .
\end{aligned} \tag{3.25}$$

3.B Derivatives of Localized LiRaM LVQ

Now we describe the derivatives of the LLiRaM LVQ scheme for one presented training example $\boldsymbol{x}^i$ with respect to the prototypes $\boldsymbol{w}^L$, the transformation matrix $\Omega \in \mathbb{R}^{M\times N}$ and the localized dissimilarities denoted by $\Psi^L \in \mathbb{R}^{M\times M}$ with $L \in \{J, K\}$. We assume the quantities of the cost function Eq. (2.14) correspond to $d_J^\Lambda = d_J^{\Psi^J}(\boldsymbol{x}^i, \boldsymbol{w}^J)$ and $d_K^\Lambda = d_K^{\Psi^K}(\boldsymbol{x}^i, \boldsymbol{w}^K)$ using the distance measure defined in Eq. (3.7). The derivative with respect to the prototypes is given by:

$$d_L^{\Psi^L}(\boldsymbol{x}^i, \boldsymbol{w}^L) = \sum_j^N \sum_k^M \sum_l^M \sum_m^M \sum_n^N (x_j^i - w_j^L)\Omega_{kj}\Psi_{lk}^L \Psi_{lm}^L \Omega_{mn}(x_n^i - w_n^L) \tag{3.26}$$

$$\frac{\partial E_{\mathrm{GMLVQ}}}{\partial \boldsymbol{w}^L} = \frac{\Phi(\mu^i)}{\partial \mu^i} \cdot \frac{\partial \mu^i}{\partial d_L^{\Psi^L}} \cdot \frac{\partial d_L^{\Psi^L}}{\partial \boldsymbol{w}^L} \tag{3.27}$$

$$\frac{\partial \mu^i}{\partial d_J^{\Psi^J}} = \gamma_\Psi^J = \frac{2 d_K^{\Psi^K}}{(d_J^{\Psi^J} + d_K^{\Psi^K})^2} \tag{3.28}$$

$$\frac{\partial \mu^i}{\partial d_K^{\Psi^K}} = \gamma_\Psi^K = \frac{-2 d_J^{\Psi^J}}{(d_J^{\Psi^J} + d_K^{\Psi^K})^2} \tag{3.29}$$

$$\frac{\partial d_L^{\Psi^L}}{\partial w_r^L} = -2 \sum_k^M \sum_l^M \sum_m^M \sum_n^N \Omega_{kr}\Psi_{lk}^L \Psi_{lm}^L \Omega_{mn}(x_n^i - w_n^L)$$

$$\frac{\partial d_L^{\Psi^L}}{\partial \boldsymbol{w}^L} = -2\Omega^\top \Psi^{L\top} \Psi^L \Omega(\boldsymbol{x}^i - \boldsymbol{w}^L) \tag{3.30}$$

The derivative with respect to the matrices is given by:

$$\frac{\partial E_{\mathrm{GMLVQ}}}{\partial \Omega} = \frac{\Phi(\mu^i)}{\partial \mu^i} \cdot \frac{\partial \mu^i}{\partial \Omega} = \Phi' \cdot \left(\gamma_\Psi^J \cdot \frac{\partial d_J^{\Psi^J}}{\partial \Omega} + \gamma_\Psi^K \cdot \frac{\partial d_K^{\Psi^K}}{\partial \Omega} \right) \tag{3.31}$$

$$\frac{\partial d_L^{\Psi^L}}{\partial \Omega_{mn}} = 2 \sum_j^N \sum_k^M \sum_l^M (x_n - w_n^L)\Psi_{kl}^L \Psi_{km}^L \Omega_{lj}(x_j - w_j^L) \tag{3.32}$$

$$\frac{\partial d_L^{\Psi^L}}{\partial \Omega} = 2 \cdot \Psi^{L\top} \Psi^L \Omega(\boldsymbol{x} - \boldsymbol{w}^L)(\boldsymbol{x} - \boldsymbol{w}^L)^\top \tag{3.33}$$

$$\frac{\partial E_{\mathrm{GMLVQ}}}{\partial \Psi^L} = \frac{\Phi(\mu^i)}{\partial \mu^i} \cdot \frac{\partial \mu^i}{\partial \Psi^L} = \Phi' \cdot \gamma_\Psi^L \cdot \frac{\partial d_L^{\Psi^L}}{\partial \Psi^L} \tag{3.34}$$

$$\frac{\partial d_L^{\Psi^L}}{\partial \Psi_{mn}^L} = 2 \sum_j^N \sum_k^N \sum_l^M (x_k - w_k^L)\Omega_{nk}(x_j - w_j^L)\Psi_{ml}^L \Omega_{lj} \tag{3.35}$$

$$\frac{\partial d_L^{\Psi^L}}{\partial \Psi^L} = 2 \cdot \Psi^L \left(\Omega(\boldsymbol{x}^i - \boldsymbol{w}^L)(\boldsymbol{x}^i - \boldsymbol{w}^L)^\top \right) \Omega^\top \tag{3.36}$$

Published as:

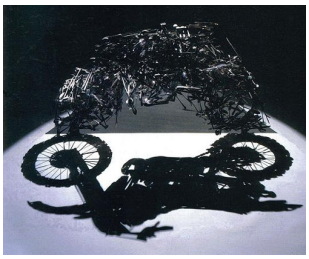
K. Bunte, M. Biehl, M.-F. Jonkman and N. Petkov – *"Learning Effective Color Features for Content Based Image Retrieval in Dermatology"*, Pattern Recognition, vol. 44, no. 9, pp. 1892–1902, 2011.

K. Bunte, M. Biehl, N. Petkov and M.-F. Jonkman – *"Adaptive Metrics for Content Based Image Retrieval in Dermatology"*, in Proc. of ESANN, pp. 129–134, Bruges, Belgium, April 2009.

Chapter 4

Adaptive Metrics for Content based Image Retrieval in Dermatology

"Lunch with a Helmet on"
848 welded forks and spoons that cast a shadow of form.



Shigeo Fukuda 1987

Abstract

In this chapter we investigate the extraction of effective color features for a Content Based Image Retrieval application in dermatology. Effectiveness is measured by the rate of correct retrieval of images from four color classes of skin lesions. We employ and compare two different methods to learn favorable feature representations for this special application: Limited Rank Matrix LVQ and the Large Margin Nearest Neighbor approach. Both methods use labeled training data and provide a discriminant linear transformation of the original features, potentially to a lower dimensional space. The extracted color features are used to retrieve images from a database by a k-Nearest Neighbor search. We perform a comparison of retrieval rates achieved with extracted and original features for eight different standard color spaces and observed significant improvements in each of them. LiRaM LVQ and the computationally more expensive LMNN give comparable results for large values of the method parameter κ of LMNN ($\kappa \geqslant 25$) while LiRaM LVQ outperforms LMNN for smaller values of κ. We conclude that feature extraction by LiRaM LVQ leads to considerable improvement in color-based retrieval of dermatologic images.

4.1 Introduction

In the last decades the availability of digital images produced by scientific, educational, medical, industrial and other applications has increased dramatically. Thus, the management of the expanding visual information has become a challenging task. Since the 1990's Content Based Image Retrieval (CBIR) is a rapidly advancing research area, which uses visual content to search images from large databases according to the user's interest (Smeulders et al. 2000, Müller et al. 2004, Lehmann et al. 2004, Datta et al. 2005, Min and Cheng 2009, Giacinto and Roli 2004, Torres et al. 2009, Jain and Vailaya 1996). A typical CBIR system extracts visual information from an image and converts it internally to a multidimensional feature vector representation. For retrieval, the dissimilarities (distances) between the feature vector of a query image and the feature vectors of the images in the database are computed. Then, the database images most similar to the query are presented to the user. CBIR may especially be interesting in the field of computer aided diagnostics when it is partly based on images. An intelligent pre-selection of images with a trained system might help a medical doctor to efficiently search for patients, who had problems similar to the actual case.

The visual content of an image can be described by color, texture, shape or spatial relationship. A good visual content descriptor should be insensitive to the specific imaging process, e.g. invariant under changes of illumination. The prevalent visual content for image retrieval is color. Frequently used color descriptors are color moments, histograms, coherence vectors and correlograms (Jau-Ling and Ling-Hwei 2002, Pass et al. 1996). Before a color descriptor can be selected, the underlying color space has to be specified. There are many different color spaces available, which may be beneficial in different application domains. The color representations most commonly used in electronic systems are RGB and CIE-XYZ. CIE-XYZ and the related CIE-Lab and CIE-Luv are designed to match human perception. In (Terrillon and Akamatsu 2000) the authors argue, that normalized TSL (Tint, Saturation, Lightness) is superior to other color spaces for skin modeling with a unimodal Gaussian joint probability density function. The color space YCrCb is adjusted for efficient image compression, but the transformation simplicity and explicit separation of luminance and chrominance components appear attractive for skin color modeling (Phung et al. 2002, Zarit et al. 1999, Chai and Bouzerdoum 2000). Surveys on color spaces and their use can be found in (Terrillon and Akamatsu 2000, Vezhnevets et al. 2003). We are not aware of a general rule for the choice of the color space and the representation might follow the users preference. So we decided to investigate eight different color spaces, which are commonly used and may be useful for the task at hand.

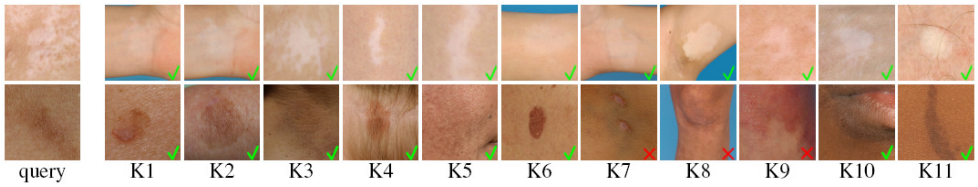


Figure 4.1: Two example retrievals of the 11 most similar images for a given query image. The first image in a row is the query image, followed by the images returned from the retrieval system (Bosman et al. 2010). The green tick marks images with the same class label like the query.

Color is an important attribute for primary skin efflorescences (Bolognia et al. 2007). Color features have proven beneficial in many applications and medical sciences, especially for the recognition of skin regions (Felice et al. 2002, Terrillon and Akamatsu 2000, Vezhnevets et al. 2003, Takiwaki 1998, Shin et al. 2002, Kjeldsen and Kender 1996, Sobottka and Pitas 1996, Phung et al. 2002, Zarit et al. 1999, Kakumanu et al. 2007) or the classification of skin cancer (Schmid-Saugeona et al. 2003, Voigt and Classen 2002, Blum et al. 2004, Hoffmann et al. 2003, Cheng et al. 2008, Umbaugh et al. 1992). A dermatologist might be interested in pictures of similar skin lesions in comparison to an actual case to verify the diagnosis or confer with similar symptoms. This can be interpreted as a problem of CBIR. The authors of (Bosman et al. 2010) study the use of color features and the effectiveness of different color spaces in this context. They conclude that the representation of an image by the difference in the average color of healthy and lesion skin gives better results than the explicit use of the pair of colors. Fig. 4.1 shows two example retrievals for a CBIR system in the field of skin lesion comparison in Dermatology. In (Bosman et al. 2010), the best results were achieved with the CIE-Lab color representation.

Of course, it is possible that the use of a combination of a cyclic distance measure in the case of color spaces containing a "hue"-descriptor might lead to superior results. We will address this interesting questions in further studies. Since the difference of two color values is a special case of a linear transformation, the question arises whether better results can be achieved by more general linear transformations. One well known technique to achieve a linear projection of feature vectors to a subspace which minimizes the overlap between different classes is Linear Discriminant Analysis (LDA) (Duda et al. 2000). In this paper we employed and compared two different recent techniques, which are able to find discriminant feature transformations based on a supervised training procedure. The Large Margin Nearest Neighbor (LMNN) (Weinberger et al. 2006) (see section 2.5) approach has the advantage that it is based on a convex cost function, so it returns the global op-

timum for the current configuration of training data and parameters. The Limited Rank Matrix LVQ (LiRaM LVQ) (Schneider et al. 2009a, Schneider et al. 2009b, Bunte et al. 2008, Schneider et al. 2008) (see Chapter 3) on the other hand follows a stochastic gradient descent procedure and may get stuck in local minima, but it has the advantage of low computational costs. Both algorithms are available in general form and turned out to be effective classifiers in many applications. In our real world example application of CBIR in Dermatology, the LiRaM LVQ approach turned out to be quite robust concerning the initialization and parameter setting. With comparably low computational costs it leads to similar or better results than the LMNN approach with optimal parameter setting on most color spaces discovered. We improve the correct retrieval rate in CBIR of dermatological images significantly by applying adaptive linear transformations.

The main aim of this work (Bunte, Biehl, Jonkman and Petkov 2011) is to demonstrate in terms of a real world example, that an adaptive, i.e. data driven transformation of original color features can improve the retrieval performance of a CBIR system significantly. We concentrate on the performance enhancement achieved by using the most basic, easy and fast acquirable set of important features for the problem at hand, i.e. color information only. In Section 4.2 we explain the real world data set and the feature extraction process. Afterwards, we discuss the results in Section 4.3 and conclude in Section 4.4.

4.2 Methodology

This work is based on the scientific findings of (Bosman et al. 2010). It has been shown, that a three-dimensional feature vector constructed from the difference between the color values of healthy and lesion skin yields better performance then using the six-dimensional feature vector of the colors itself. Since the difference features are acquired by a simple fixed linear transformation A the question arises if the CBIR system can improve even further using an arbitrary transformation. Therefore we compare two supervised adaptive distance techniques, namely LiRaM LVQ and LMNN, which are able to provide discriminative transformations of the feature space used for CBIR. An illustration of the Methodology is shown in Figure 4.2.

4.2.1 Data set and feature extraction

We analyze images from a database maintained at the Department of Dermatology of the University of Groningen. At the time of this study it consisted of 47621 images from 11361 patient sessions, the number of images grows by about 5000 per year. Clinical images are obtained under standard light conditions and do not require

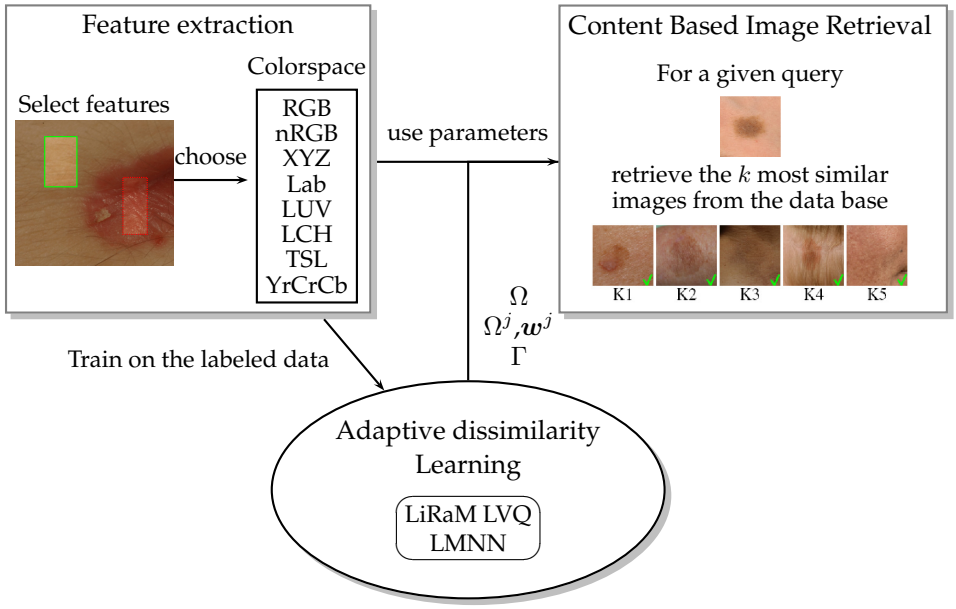


Figure 4.2: Methodology overview for the proposed CBIR system.

(a) Red (b) White (c) Blue (d) Brown

Figure 4.3: Example images of the four skin lesion classes from (Bosman et al. 2010).

further calibration. A subset of 211 images was provided and manually labeled by a dermatologist, who assigned each image to one of four classes of lesions. For better readability we refer to these classes as “red”, “white”, “blue” and “brown”, see Fig. 4.3. These terms correspond to the relative tint of lesions which appear reddish, hypo-pigmented, blue or brownish on the background of the surrounding healthy skin. We consider a data set with 82, 46, 29 and 54 samples, respectively, which amounts to a total of 211 images.

Of course there are more characteristics then just color which identify the kind

of skin lesion, e.g. the shape. The consideration of other types of features will be addressed in future work, here we concentrate on the quality the most basic set of features is able to achieve. In this particular problem color seems to be a suitable indicator for the skin lesion classes. The complete data set also contains other skin lesions, but in this study we restrict ourselves to the consideration of the above mentioned classes. Here, emphasis is not on the classification performance itself. It serves as a basis for improving the retrieval system and the supervised training yields a suitable distance measure. Further studies should address additional features, more general skin lesion classes and the handling of unknown classes.

The original images were not preprocessed. For each image a region of lesion and a region of healthy skin are manually selected and for each of them the average color values are computed (see Fig. 4.4). Hence, the extracted data contains three color components for each of the two regions, resulting in a six-dimensional feature vector $\boldsymbol{x} \in \mathbb{R}^6$. As a normalization step we perform a z-transformation resulting in zero mean and unit variance features. This normalization is reasonable in the RGB color space and linear domains. In case of cyclic descriptors, like the "hue", this might not be appropriate. The combination of cyclic distances and linear dissimilarities and their normalization concerning this specific task will be addressed in future studies. Nevertheless, for the sake of comparison and completeness we show the results on different color spaces under the same conditions.

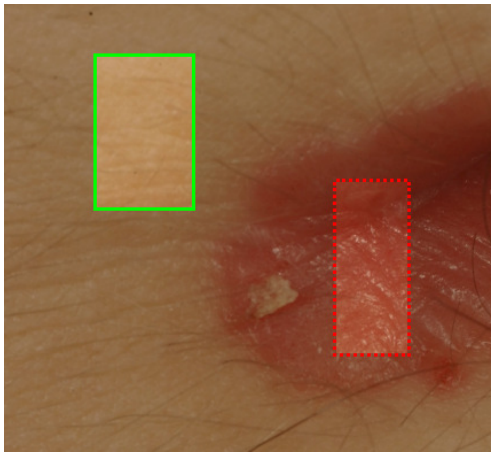


Figure 4.4: Feature extraction: a representative region of healthy skin (green) and lesion skin (red) were manually selected. The average colors of these regions are combined in a six-dim. feature vector.

4.2.2 Feature transformation obtained by LiRaM LVQ

In order to obtain discriminative representations of the data we employ LiRaM LVQ technique, which is explained in Section 3.2. Following Eq. (2.13) we transform the features into a discriminative space $\boldsymbol{\xi} = \Omega\boldsymbol{x}$, which is then used in the CBIR system. The results of the LiRaM LVQ algorithm may display a dependence on the initial

state of the matrix Ω in the training. Hence, we present results on average over several random initial configurations. For the training we employ the following cross validation procedure: The data set is split in ten disjoint subsets with approximately the same composition of classes. The union of nine subsets is used to determine the transformation matrix Ω for the vectors of the remaining subset. In this way, the matrix Ω which is applied to a given feature vector from the set is obtained without using that feature vector. This procedure is repeated ten times, once for every possible selection of the subset for which Ω is determined. In addition we repeat each training process for ten different random initializations of the LiRaM LVQ algorithm, resulting in 100 runs.

We start the matrix learning after $t_M = 50$ of altogether 500 epochs t and apply a learning rate schedule shown in Eqs. (3.5) and (3.6), which has proven advantageous in many implementations of relevance learning. In our experiments we chose $\tau_1^{\text{start}} = 0.01$, $\Delta\tau_1 = \Delta\tau_2 = 0.0001$ and $\tau_2^{\text{start}} = 0.001$, we do not perform an optimization of these parameters concerning the retrieval rates. In our experiments we use four prototypes (one per class) and their initial positions $\boldsymbol{w}^i(t = 0)$ are determined as the mean over a random selection of $1/3$ of the available feature vectors in class $c(\boldsymbol{w}^i)$ with small random deviation. Hence, prototypes are initially close to the class-conditional means in the training data, but with small deviations due to the random sampling. This has the advantage that in the case of more prototypes it is ensured that they are not initialized on exactly the same position. Relevance initialization is done by generating independent uniform random numbers $\Omega_{ij} \in [-1, 1]$ and subsequent normalization Eq. (2.21). Performing independent runs with random initialization and subsequent normalization prevents that single features are favored by unlucky initialization. In the experiments we consider matrices $\Omega \in \mathbb{R}^{3\times 6}$, which transform the original six-dimensional feature vectors $\boldsymbol{x}$ into a three-dimensional space. More dimensions do not increase the performance significantly, but using less than three caused decreasing retrieval rates. Furthermore, with three dimensions we can directly compare to earlier experiments.

The Localized GMLVQ (LGMLVQ) (see Algorithm 2.4) using localized dissimilarities Eq. (2.22) is trained under the same conditions and learning rate schedules, adapting four matrices $\Omega^j \in \mathbb{R}^{3\times 6}$ together with their associated prototypes $\boldsymbol{w}^j$ in the supervised training process.

For each subset D^s, $s = 1, \ldots, 10$, of the data set $\mathcal{X}$ we perform 10 runs over random initializations $i = 1, \ldots, 10$. For every image $\boldsymbol{x}^j$ with $j = 1, \ldots, 211$ from the data set we compute the correct retrieval rate by means of the k nearest neighbors within $\mathcal{X}\backslash\{\boldsymbol{x}^j\}$. Therefore, we apply for each initialization i the transformation Ω^{si} or $\Omega^{l,si}$ in the localized version, which was learned without the samples $\boldsymbol{x} \in D^s$, and obtain a retrieval rate r_j^i for the query $\boldsymbol{x}_j \in D^s$. Thus we get for every initialization

i a mean retrieval rate $\bar{r}^i = \frac{1}{211}\sum_{j=1}^{211} r_j^i$. As an overall estimate of the performance we determine the total mean rate $r = \frac{1}{10}\sum_i^{10} \bar{r}^i$. The variability with respect to initialization is quantified by the standard deviation

$$\sigma_{\text{init}} = \left(\frac{1}{9}\sum_{i=1}^{10}(\bar{r}^i - r)^2\right)^{\frac{1}{2}} . \tag{4.1}$$

In order to quantify the variation of the data set we evaluate the mean retrieval rate of every image $\bar{r}_j = \frac{1}{10}\sum_{i=1}^{10} r_j^i$ and the corresponding standard error of mean:

$$\epsilon_{\text{data}} = \left(\frac{1}{210}\sum_{j=1}^{211}(\bar{r}_j - r)^2\right)^{\frac{1}{2}} \cdot 211^{-\frac{1}{2}}. \tag{4.2}$$

With the original features there is no training process involved and ϵ_{data} in Eq. (4.2) is computed simultaneously with the retrieval rate r_j of every image replacing $\bar{r}_j$.

4.2.3 Feature transformation obtained by LMNN

We also perform the LMNN method (Weinberger et al. 2006) explained in Section 2.5 to acquire discriminant transformations of the feature space. The results presented in the following section were produced with the available code[1] using default parameters except for the number of target neighbors κ, which varies in our experiments from 1 to 25 and the matrix $\Gamma = \Upsilon^\top\Upsilon$ Eq. (2.30) decomposed by $\Upsilon \in \mathbb{R}^{3\times 6}$ initialized with elements randomly drawn from the interval $[-1, 1]$. For a fair comparison, LMNN and LiRaM LVQ are applied to the same subsets D^s of training data and performance is evaluated on the same footing as explained before.

4.2.4 Canonical representations

Note that the transformation matrix Ω obtained by LiRaM LVQ and Υ in LMNN are not uniquely determined: For instance, the distance measure is invariant under rotations in the feature space. Thus, the training process can yield different transformation matrices depending on the (random) initialization of the training process. We identify unique transformations $\widehat{\Omega}$ and $\widehat{\Upsilon}$ by decomposing $\Lambda = \Omega^\top\Omega$ and $\Gamma = \Upsilon^\top\Upsilon$ in a canonical way based on the sorted eigenvectors $\boldsymbol{v}^j$ following Eq. (3.2):

$$\widehat{\Upsilon}, \widehat{\Omega} = \left(\left[\sqrt{\lambda_1}\,\boldsymbol{v}^1, \sqrt{\lambda_2}\,\boldsymbol{v}^2, \cdots, \sqrt{\lambda_M}\,\boldsymbol{v}^M\right]\right)^\top \in \mathbb{R}^{M\times N} \quad . \tag{4.3}$$

[1] www.cse.wustl.edu/∼kilian/code/code.html (last visited September 2010)

This canonical representation does not alter the retrieval system and it allows direct comparison of the transformations $\widehat{\Omega}$ and $\widehat{\Upsilon}$.

It is not obvious how to extend the LMNN scheme for a comparison with the use of local matrices Ω^j like in the LiRaM LVQ. Localized transformations could heuristically be put on top of the LMNN scheme by forcing a separation of the feature space, e.g. based on the class information. Since the LMNN scheme computes distances within the feature space it is not clear which distance should be used when comparing two sampes of two different classes:

$$d^{\Upsilon^{y^i}}(\boldsymbol{x}^i, \boldsymbol{x}^j) \neq d^{\Upsilon^{y^j}}(\boldsymbol{x}^j, \boldsymbol{x}^i) \text{ assuming } \Upsilon^{y^i} \neq \Upsilon^{y^j} \quad . \tag{4.4}$$

LVQ, on the other hand, contains a quantization process within the learning procedure, which makes localized transformations within the receptive fields a very natural and easy extension. The distances are always computed with respect to the prototypes, not the samples itself.

4.2.5 Retrieval test

As a performance measure for CBIR we use the average correct retrieval rate, also referred to as precision. It is defined as the percentage of k-Nearest Neighbors (k-NNs) that belong to the same category as a query image. We determine for each image its k-NNs in the entire data set using the Euclidean distance measure. For comparison, we do this both in the original feature space $\mathcal{X}$ and in the transformed feature space $\mathcal{E} = B\boldsymbol{x}$ with $B \in \{\Omega, \Upsilon\}$. Note that in our evaluation for a given query image, the transformation matrices Ω, Υ and Ω^j have been determined from subsets which do not contain the query.

Using the Generalized Matrix LVQ (GMLVQ) approach the training process optimizes j localized transformations Ω^j corresponding to the classification task. We involve this information by projecting every feature vector $\boldsymbol{x}$ with the transformation Ω^J corresponding to the nearest prototype $\boldsymbol{w}^J$ with $d^{\Lambda^J}(\boldsymbol{w}^J, \boldsymbol{x}) < d^{\Lambda^l}(\boldsymbol{w}^l, \boldsymbol{x})$ $\forall J \neq l$ resulting in local linear projections for different areas of the feature space.

Section 4.3 presents and compares the resulting retrieval rates as average over all images. Furthermore, the standard error of the performance with the actual query image and its dependence on the initialization of LiRaM LVQ are discussed.

4.2.6 Color spaces

We explore the retrieval rates for eight different color representations separately. The different color spaces vary, as already mentioned, with respect to their usefulness in different applications. Possible motivations for the choice of a particular

Table 4.1: Overview over some color spaces compared for their use in CBIR.

Color space	chosen for:
RGB	widespread use
normalized RGB	invariance (under certain assumptions) to changes of surface orientation with respect to the light source (Skarbek and Koschan 1994)
TSL	successful application in skin detection (Terrillon and Akamatsu 2000)
CIE-XYZ	role as the basis for CIE-Lab and CIE-Luv
CIE-Lab	perceptual relevance and relation to melanin and hemoglobin (Takiwaki 1998)
CIE-Luv & CIE-Lch	perceptual relevance
YCrCb	simplicity and explicit separation of luminance and chrominance components (Phung et al. 2002, Zarit et al. 1999) and popularity in skin detection applications (Kakumanu et al. 2007)

color space are summarized in Table 4.1. Despite the potential difficulty rising from the cyclic representation of the "Hue" component of the TSL color space and its relatives HSV and HSL, for completeness, we investigate its behavior for our application task in terms of one example, namely TSL.

4.3 Results

4.3.1 Retrieval rates

In this Section we summarize the retrieval results for the different color representations using transformed features from LMNN, LiRaM LVQ and GMLVQ. We compare them with those obtained in the original feature spaces and with the difference features from (Bosman et al. 2010) obtained with the transformation $\boldsymbol{\xi} = A\boldsymbol{x}$ with:

$$A = \begin{pmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{pmatrix} . \tag{4.5}$$

The overall mean rates r obtained with LiRaM LVQ and $\Omega \in \mathbb{R}^{3\times 6}$ are displayed in Fig. 4.5 for each color space as a function of the number k, i. e. the number of pictures the CBIR system returns to the user. The best correct retrieval rates for this

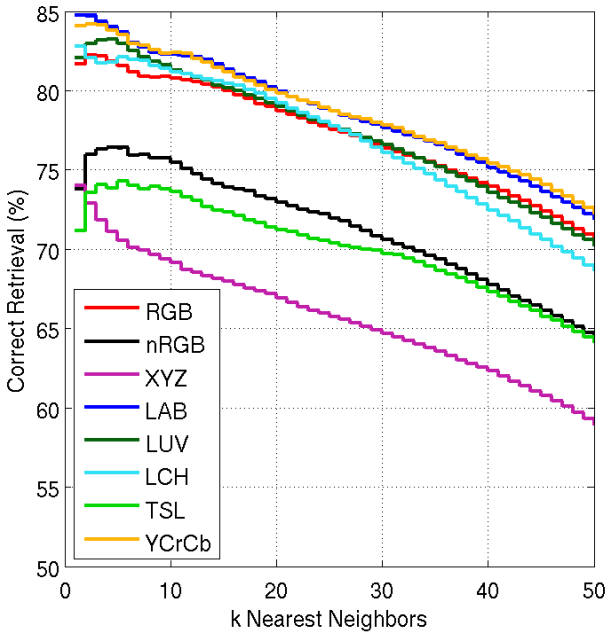


Figure 4.5: Mean correct retrieval rates obtained with the LiRaM LVQ transformed data as a function of the number k of retrieved images for eight color spaces.

algorithm are achieved with the color spaces YCrCb (82.3%), CIE-Lab (82.2%), CIE-Lch (81.1%), CIE-Luv (81.0%) and RGB (80.7%) where the numbers correspond to the example case $k = 11$. All other color representations yield by far lower performances with rates between 68.7% and 75.0%. We chose the example case of 11 returned images for the quantitative analysis to be able to compare to earlier studies (Bosman et al. 2010) and because it seems a reasonable large number suggested by the doctor. Of course the system is able to return as many similar images as the data base contains and the user wishes to see.

Fig. 4.6 shows a comparison of the correct retrieval rates based on the original features (red lines), the difference features from (Bosman et al. 2010) (green lines) and the transformed data (blue and black lines) as a function of the neighborhood size k of the retrieval system. The gray shaded areas mark the standard error of mean ϵ_{data}, while the blue shaded area corresponds to σ_{init} of the LiRaM LVQ. Note that the latter is, of course, absent in the results based on original features and difference features, as no training process is involved and also absent in the results

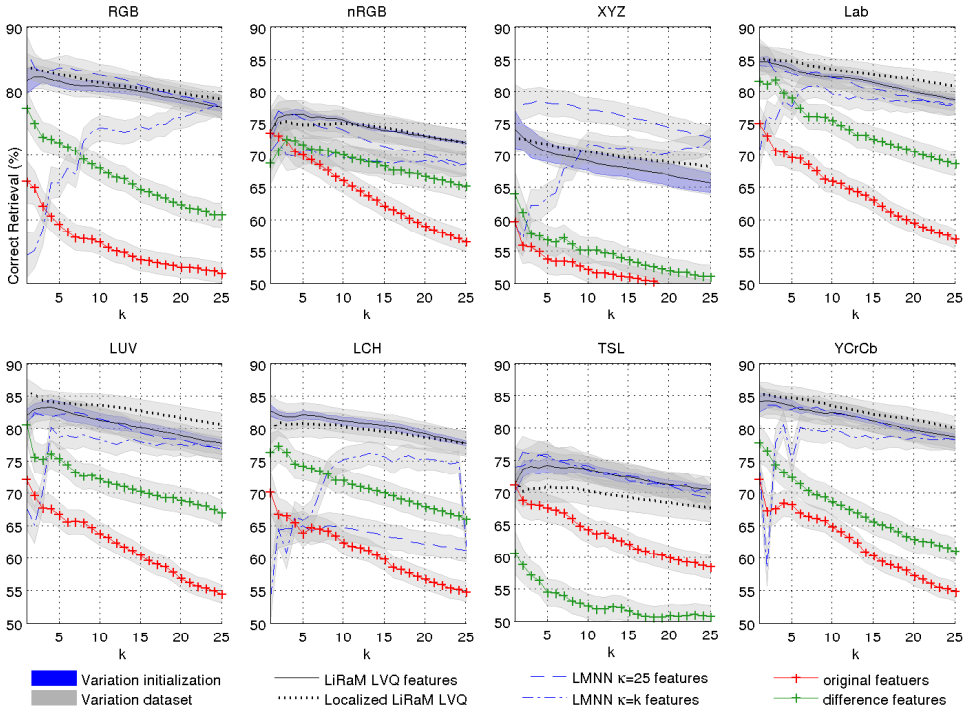


Figure 4.6: Comparison of correct retrieval rates dependent on the number of nearest neighbors k for each color space. The red lines denote the mean retrieval rates on the original feature space, the green line stands for the difference features from (Bosman et al. 2010), whereas the blue and black lines shows the mean results on the transformed feature spaces. The blue shaded areas indicates the standard deviation due to the random initializations σ_{init} in LiRaM LVQ.

coming from LMNN, because it finds the global optimum for a given parameter set, independent of the initial state. The variation due to initialization of the GMLVQ is not displayed; it is comparable to the variation in the global version. We set the parameter κ of the LMNN approach equal to the neighborhood k of the retrieval system and, in addition, we consider $\kappa = 25$. The latter is close to the size of the smallest class in the data set, "blue" (c), with 29 examples. For $\kappa = 25$ the retrieval performances of LMNN and LiRaM LVQ are comparable which is also reflected in the fact that the obtained matrices $\hat{\Omega}$ and $\hat{\Upsilon}$ are very similar, cf. Fig. 4.7 and Fig. 4.8. Smaller values for κ reduce the computational effort of the optimization at the expense of performance.

LGMLVQ achieves the best correct retrieval rate for the most suitable color spaces: Lab and YCrCb. However, the performance boost compared to the other methods is only moderate. In TSL, GMLVQ is even outperformed by the simpler techniques based on global measures. These findings suggest that the latter already extract the most important information from the original color features. Furthermore, TSL is cyclic represented by the angle of color components, which may cause instabilities for naive distance computation. We suggest the performance drop of the difference features in comparison to the use of the original features is a consequence of the Hue representation in TSL and its relatives HSL and HSU were we observed the same effect. However, the adaptive distance is able to compensate for this effect and still yields a boost of performance also in these color spaces.

In most of the color spaces, including RGB, the LiRaM LVQ result is not very sensitive to initialization, as indicated by relatively small standard deviations $\sigma_{\text{init}} \leqslant 2\%$. The XYZ color representation display the largest dependence on initialization with $\sigma_{\text{init}} > 2.7\%$. The variation with the data set is approximately the same in original and transformed feature spaces. This variability is not an effect of the LiRaM LVQ training but is characteristic of the data set itself. In the case of the LMNN optimization, we observe that the use of an adaptive transformation increases the mean retrieval rate r significantly for all color spaces, for every choice of k and appropriate κ. The best results are obtained with CIE-Lab ($72\% < r < 85\%$) and YCrCb ($72\% < r < 84\%$). It is interesting to note that the popular RGB representation exhibits comparable performance ($70\% < r < 82\%$) in the transformed feature space. Thus, we achieve an improvement between 10% and 27% when employing an adaptive linear transformation of features.

4.3.2 Recommended transformations

Here we inspect the favorable transformations of the feature space as obtained by LiRaM LVQ and LMNN. We focus on RGB as the by far most frequently used color space and on CIE-Lab because of its excellent retrieval performance.

Global transformations

We observe that the obtained distance measure represented by Λ depends only weakly on the initialization of LiRaM LVQ. However, a continuum of matrices Ω satisfies $\Omega^\top \Omega = \Lambda$ and, in this sense, the actual outcome Ω of the training process can vary widely. Thus, the canonical representation $\widehat{\Omega}$ Eq. (4.3) is averaged over all training runs. The mean transformation is explicitly given for RGB in Eq. (4.6) and visualized in Fig. 4.7. The standard deviation concerning the random initialization of each component lies between 0.01 to 0.03 for $\widehat{\Omega}_{\text{RGB}}$. Each row of the matrix

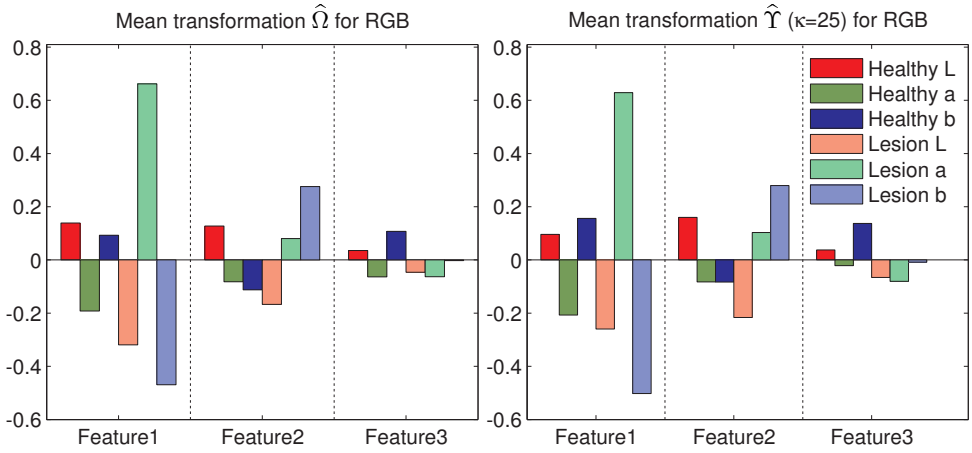


Figure 4.7: Recommendation for the transformation in RGB: (left) Multipliers that define the new features as linear combinations of the original features earned from LiRaM LVQ. (right) Multipliers earned from LMNN with $\kappa = 25$.

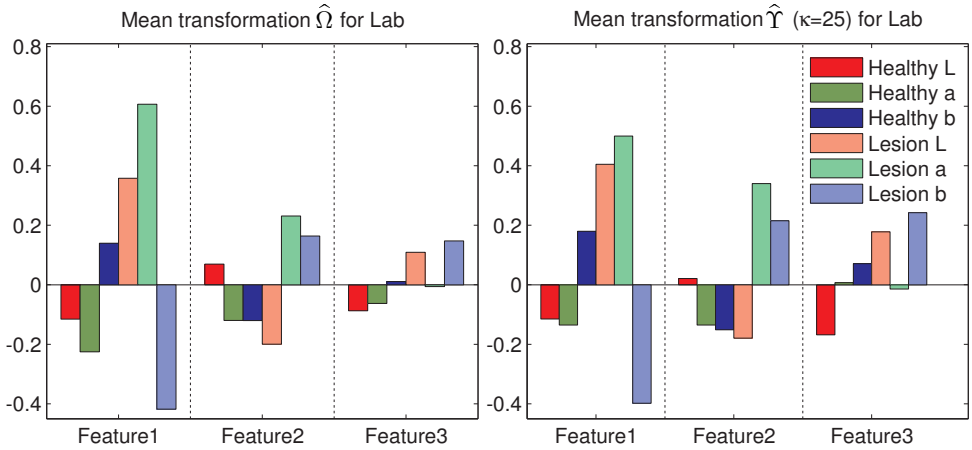


Figure 4.8: Recommendation for the transformation in CIE-Lab: (left) Multipliers that define the new features as linear combinations of the original features earned from LiRaM LVQ. (right) Multipliers earned from LMNN with $\kappa = 25$.

defines a new feature as a linear combination of the original six features.

$$\widehat{\Omega}_{\mathrm{RGB}} = \begin{pmatrix} 0.139 & -0.192 & 0.093 & -0.320 & 0.662 & -0.469 \\ 0.127 & -0.082 & -0.112 & -0.167 & 0.080 & 0.276 \\ 0.036 & -0.064 & 0.108 & -0.047 & -0.063 & -0.002 \end{pmatrix} \tag{4.6}$$

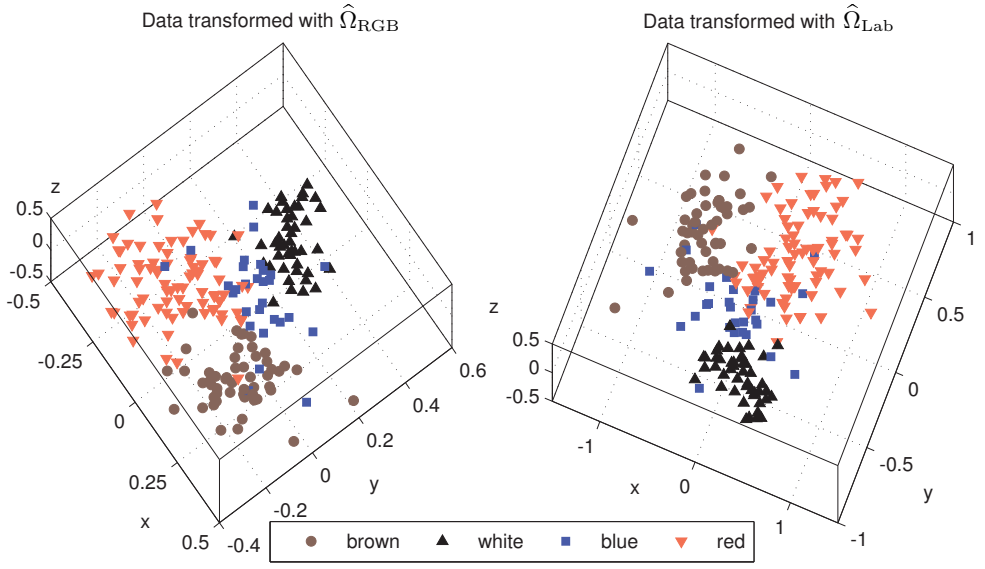


Figure 4.9: The resulting 3D visualizations of the skin cancer data set transformed from the RGB and LAB color space with $\widehat{\Omega}_{\mathrm{RGB}}$ (left panel) and $\widehat{\Omega}_{\mathrm{Lab}}$ (right panel).

We observe, that the absolute weights corresponding to skin lesions (columns 4,5,6) are typically 1-2 times larger than the coefficients assigned to the healthy skin features (columns 1,2,3). In general, the corresponding coefficients for lesion and healthy skin features are of opposite sign. Hence, the transformed features correspond to weighted differences of the lesion and healthy skin color values. Eq. (4.7) denotes explicitly the mean transformation $\widehat{\Omega}_{\mathrm{Lab}}$ for CIE-Lab; it is visualized in Fig. 4.8:

$$\widehat{\Omega}_{\mathrm{Lab}} = \begin{pmatrix} -0.115 & -0.225 & 0.140 & 0.358 & 0.606 & -0.418 \\ 0.069 & -0.120 & -0.120 & -0.200 & 0.231 & 0.164 \\ -0.087 & -0.063 & 0.011 & 0.109 & -0.006 & 0.147 \end{pmatrix} . \tag{4.7}$$

The above discussed properties of Ω_{RGB} persist also in the transformation of CIE-Lab feature vectors. The standard deviations for the mean transformation vary from 0.01 and 0.06 for the random initializations.

The resulting 3D visualizations of the data set with the mean canonical transformations $\widehat{\Omega}$ using the RGB and LAB color representation are shown in Fig. 4.9. It can be seen that the classes for ”white“, ”red“ and ”brown“ skin cancer build a nicely separable data cloud respectively, whereas the class ”blue“ lays between the others and overlaps. With more training samples especially of the difficult class the data set might be even better separable by supervised adaptive dissimilarity learning.

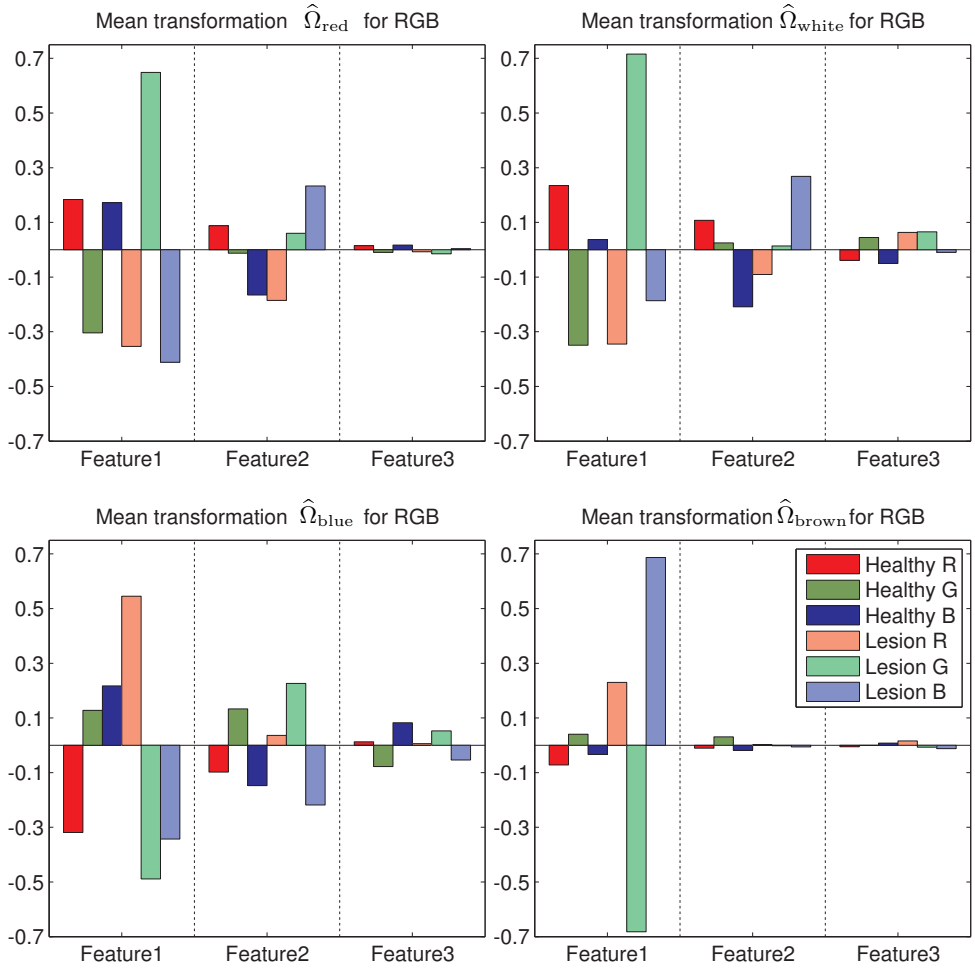


Figure 4.10: Local Matrices for RGB corresponding to one prototype of each class.

Local transformations

Also with the localized matrices the above discussed properties persist. For the local feature transformation the prototypes are necessary and define the area of the original feature space, where their transformation is valid. So the samples are transformed with the transformation attached to the nearest prototype $\boldsymbol{w}^J$:

$$\boldsymbol{\xi} = \Omega^J \boldsymbol{x} \text{ with } d^{\Lambda^J}(\boldsymbol{w}^J, \boldsymbol{x}) = \min_j d^{\Lambda^j}(\boldsymbol{w}^j, \boldsymbol{x}) \ . \tag{4.8}$$

The mean canonical representations of the local matrices for RGB are shown in Fig. 4.10. Note that the definition in Eq. (4.8) is only valid in the neighborhood of the

corresponding prototype. At the borders of the Voronoi cell of each prototype this definition may be inappropriate. In general it is possible to combine the local linear patches in a global nonlinear way by charting (Bunte, Hammer, Wismüller and Biehl 2010, Brand 2002) or Local Linear Coordination (LLC) (Teh and Roweis 2003). It can be seen that some class-wise transformations seems to be already well discriminating with one or two features, for example the matrices for the "brown" and "red" class of skin lesions. However, for the class of white and bluish appearing skin lesions also the third feature shows a contribution to the transformation. It would have been possible to have class-wise different target spaces for two and one dimension in respective transformations, but for reasons of consistency and for comparison purpose we chose the target dimension to be the same for every class.

In summary, our findings support the basic idea of using differences of color features presented in (Bosman et al. 2010). We have shown, however, that generalizing this concept by introducing adaptive coefficients improves the retrieval performance significantly for this supervised problem.

4.4 Summary and conclusion

In this chapter we show the usefulness of adaptive distances and corresponding feature space transformations on a real world example application. We observe that CBIR on color is a powerful tool for analysis of dermatological image databases. Previously unnoticed color similarities may give new insight into the correlations between and within various skin diseases. We introduce discriminative color descriptors which are obtained by LiRaM LVQ and LMNN during supervised training, and we compare and evaluate their performance for CBIR of dermatological images. Starting from a 6D vector representation of images, we define three new features as linear combinations of the original six color components of healthy and lesion skin. The linear combinations are determined by LiRaM LVQ in a training process which is guided by classification performance and yields a discriminative representation of the feature space. With new features we achieve considerable improvement of retrieval results in all eight color spaces that we studied. In the five best color spaces (YCrCb, CIE-Lab, CIE-Lch, CIE-Luv and RGB) the increase of the correct retrieval rate is between 10% and 27% in the range of $k = 1$ to $k = 25$ retrieved images in comparison to earlier studies. We conclude that adaptive dissimilarity learning is favorable independent of the choice of the actual color space. The user may decide according to his personal preference which color representation is most suitable.

The use of LMNN seems natural, since the retrieval is based on a k-NN approach. However, our investigation shows that the LiRaM LVQ approach outper-

forms LMNN if the latter takes only a relatively small number κ of neighbors into account in the training process. For larger κ the obtained metric becomes very similar to that of LiRaM LVQ and, consequently, the retrieval performances are comparable. The computational effort for LiRaM LVQ training is typically lower than that of the LMNN optimization which grows with κ. An important advantage of the LVQ approach is its potential with respect to extensions. As shown, for example, local metrics can be attached to the prototypes which are responsible for different areas of the original feature space. In the most favorable color spaces, the localized variant GMLVQ increased the retrieval rates even further.

We conclude that LiRaM LVQ is an efficient technique for the extraction of highly discriminative color features for CBIR of dermatological images. With this approach, we obtain high mean correct retrieval rates of between 84% for $k = 1$ and 79% for $k = 25$ retrieved images in the five best color spaces. For two of the color spaces, RGB and CIE-Lab, we discuss in detail the canonical linear transformations of the original six color components to three new features and showed their superiority to recently introduced approaches.

Obviously, several important extensions are possible. For instance, the automatic detection of regions of interest or the integration of shape information should be relevant in practical applications. Forthcoming studies should address, among other modifications, the use of extended original feature spaces which include, for instance, shape information.

Published as:

K. Bunte, I. Giotis, N. Petkov and M. Biehl – *"Adaptive Matrices for Color Texture Classification,"* in Proc. of 14th International Conference on Computer Analysis of Images and Patterns (CAIP), vol. 6855, Part II, pp. 489–497, Seville, Spain, August 2011.

Chapter 5

Adaptive Matrices for Color Texture Classification

Art is the imposing of a pattern on experience,
and our aesthetic enjoyment is recognition of the pattern.

Alfred North Whitehead (1861 - 1947)

Abstract

In this chapter we introduce an integrative approach towards color texture classification learned by a supervised framework. Our approach is based on the Generalized LVQ (GLVQ), extended by an adaptive distance measure which is defined in the Fourier domain and 2D Gabor filters. We evaluate the proposed technique named Color Image Analysis LVQ (CIA LVQ) on a set of color texture images and compare results with those achieved by simple gray value transformation on the color images with a comparable dissimilarity measure and the same filter bank. The features learned by CIA LVQ improve classification accuracy and they generalize much better for evaluation data previously unknown to the system.

5.1 Introduction

Texture analysis and classification are topics of particular interest mainly due to their numerous possible applications, such as medical imaging, industrial quality control and remote sensing. Despite the absence of a unique definition, texture is understood as a description of the spatial arrangement of colors or intensities in an image. A wide variety of methods for texture analysis has been already developed such as co-occurrence matrices (Haralick et al. 1973), Markov random fields (Wang and Liu 1999), autocorrelation methods (Pietikäinen et al. 2000, Ojala et al. 2002), Gabor filtering (Turner 1986, Fogel and Sagi 1989, Jain and Farrokhnia 1991, Kruizinga

and Petkov 1995, Manjunath and Ma 1996, Grigorescu et al. 2002) and wavelet decomposition (Wang et al. 1998). However, these methods mostly concern intensity images and since color information is a vector quantity the transfer of traditional methods to the color domain is not always straightforward. With regards to color texture the possible approaches can be distinguished in three categories (Palm 2004). In the parallel approach (Messer and Kittler 1999, Paschos 2000) textural features are extracted solely from the luminance plane and are used together with color features. The sequential approach (Hauta-Kasari et al. 1999) involves a quantization of the color space and subsequently the extraction of statistical features from the indexed images. The most popular among them is called the integrative approach (Jain and Healey 1998, Drimbarean and Whelan 2001, Palm 2004, Hoang et al. 2005) and is an attempt to describe texture by combining color information with the spatial relationships of image regions within each color channel and between different color channels.

We introduce a novel integrative approach towards color texture classification and recognition based on 2D Gabor filters and supervised learning (Bunte, Giotis, Petkov and Biehl 2011). Given a set of labeled color images (RGB) for training and a bank of 2D Gabor filters the goal here is to learn a transformation of a color image to a single channel (intensity) image, such that the Gabor responses of the transformed images will yield the best possible classification. Most signal processing techniques are based on insights or empirical observations from neurophysiology or optical physics. The proposed, novel approach incorporates data-driven adaptation of the system, e.g. example based learning. Furthermore, the filters used in our approach can be substituted, depending on the data domain and the task at hand. As an example we explore the use of rotation and scale invariant descriptors based on Gabor filter responses (Han and Ma 2007). We demonstrate that our novel approach yields very good generalization ability with respect to previously unknown data.

In Section 5.2 we introduce the LVQ based color texture learning method. The experiments are shown in Section 5.3 and finally we conclude in Section 5.4.

5.2 Adaptive matrices for texture classification

We consider a data set consisting of color image patches of a priorly defined size ($s \times s$) and a bank of Gabor kernels $\mathbf{G}$ with different scales and orientations. We use for both the image patches and the filter kernels their representation in the Fourier domain. After vectorizing we end up with complex data points $\boldsymbol{x}^i \in \mathbb{C}^N$ of dimension $N = s \cdot s \cdot 3$ carrying a label $y^i \in \{1, \ldots, C\}$ that belong to one of C classes. $\mathbf{G}^l \in \mathbb{C}^M$ with $M = s \cdot s$ is the vectorized kernel of the l-th filter of the bank $\mathbf{G}$. The

general form of the descriptor for a vectorized patch $\boldsymbol{v}$ given the filter bank $\mathbf{G}$ and parameterized by local transformations Ω^k can be written as $f_{\Omega^k}(\boldsymbol{v}, \mathbf{G}) : \mathbb{C} \rightarrow \mathbb{C}$. Here k corresponds to the index of the prototype $\boldsymbol{w}^k$ or the index of its class label $c(\boldsymbol{w}^k)$ for class-wise transformations. For the proposed optimization procedure it is necessary, that f_{Ω^k} is differentiable. In this contribution f_{Ω^k} corresponds to the sum of the responses of all filter kernels in $\mathbf{G}$ to the vectorized patch, thus defining the descriptor:

$$f_{\Omega^k}(\boldsymbol{v}, \mathbf{G}) : \boldsymbol{v} \rightarrow \boldsymbol{r}^k(\boldsymbol{v}) = \sum_l \Omega^k \boldsymbol{v} * \mathbf{G}^l \ , \tag{5.1}$$

where $*$ denotes the convolution. The filter bank $\mathbf{G}$ may be chosen based on the user's preference, suitable to the data and the task at hand. The vector $\boldsymbol{v}$ is defined in the data domain $\mathbb{C}^N$ and $\Omega^k \in \mathbb{C}^{M \times N}$ is the local transformation, which maps the color values to scalar, "intensity" values used for filtering. The dissimilarity measure is defined by:

$$d_{\mathbf{G}}^{\Omega^k}(\boldsymbol{x}^i, \boldsymbol{w}^k) = \parallel |\boldsymbol{r}^k(\boldsymbol{x}^i)|^2 - |\boldsymbol{r}^k(\boldsymbol{w}^k)|^2 \parallel^2 \tag{5.2}$$

and corresponds to the difference of descriptor magnitudes. This considers two patches containing the same texture pattern as similar, independent of the position where the pattern occurs within the patches.

We use the same cost function as in the original GLVQ algorithm Eq. (2.5) including the dissimilarity measure defined by Eq. (5.2):

$$E_{\mathrm{CIA}} = \sum_{i=1}^{n} \frac{d_{\mathbf{G}}^{\Omega^J} - d_{\mathbf{G}}^{\Omega^K}}{d_{\mathbf{G}}^{\Omega^J} + d_{\mathbf{G}}^{\Omega^K}}, \text{ with } d_{\mathbf{G}}^{\Omega^L} = d_{\mathbf{G}}^{\Omega^L}(\boldsymbol{x}^i, \boldsymbol{w}^L) \text{ for } L \in \{J, K\} \ . \tag{5.3}$$

We follow a stochastic gradient descent procedure and present the samples $\boldsymbol{x}^i$ of the training set sequentially and update the parameters accordingly. We will refer to this algorithm as CIA LVQ (see Algorithm 5.1). The detailed description of the derivatives $\frac{\partial E_{\mathrm{CIA}}}{\partial \boldsymbol{w}^L}$ and $\frac{\partial E_{\mathrm{CIA}}}{\partial \Omega^L}$ for $L \in \{J, K\}$ defining the learning rules can be found in Appendix 5.A. A short scheme of the method is also depicted in Fig. 5.1. In the next section we experiment with the algorithm and show its use in practice.

5.3 Experiments

In order to evaluate the usefulness of the proposed algorithm, we perform classification on patches of pictures taken from the VisTex database (VisTex 2002). Our data consists of color images with size 128×128 pixels from the groups Bark, Brick, Tile, Fabric and Food. Although in texture classification literature each such image

Algorithm 5.1 : Color Image Analysis LVQ (CIA LVQ)

1: define a filter bank $\mathbf{G}$
2: initialize the prototypes $\boldsymbol{w}^j$ and their labels $c(\boldsymbol{w}^j)$
3: initialize matrices Ω^j
4: **while** stopping criterion not reached **do**
5: randomly select a training sample $\boldsymbol{x}^i$
6: compute the distances $d_{\mathbf{G}}^{\Omega^j}(\boldsymbol{x}^i, \boldsymbol{w}^j)$ to the prototypes $\boldsymbol{w}^j$
7: determine closest correct $\boldsymbol{w}^J = \arg\min_j d_{\mathbf{G}}^{\Omega^j}(\boldsymbol{x}^i, \boldsymbol{w}^j)$ with $y^i = c(\boldsymbol{w}^J)$
and closest incorrect $\boldsymbol{w}^K = \arg\min_j d_{\mathbf{G}}^{\Omega^j}(\boldsymbol{x}^i, \boldsymbol{w}^j)$ with $y^i \neq c(\boldsymbol{w}^K)$
8: update the prototypes according to $\boldsymbol{w}^L \leftarrow \boldsymbol{w}^L - \tau_1 \cdot \frac{\partial E_{\mathrm{CIA}}}{\partial \boldsymbol{w}^L}$, $L \in \{J, K\}$
9: update the matrices according to $\Omega^L \leftarrow \Omega^L - \tau_2 \cdot \frac{\partial E_{\mathrm{CIA}}}{\partial \Omega^L}$
10: **end while**

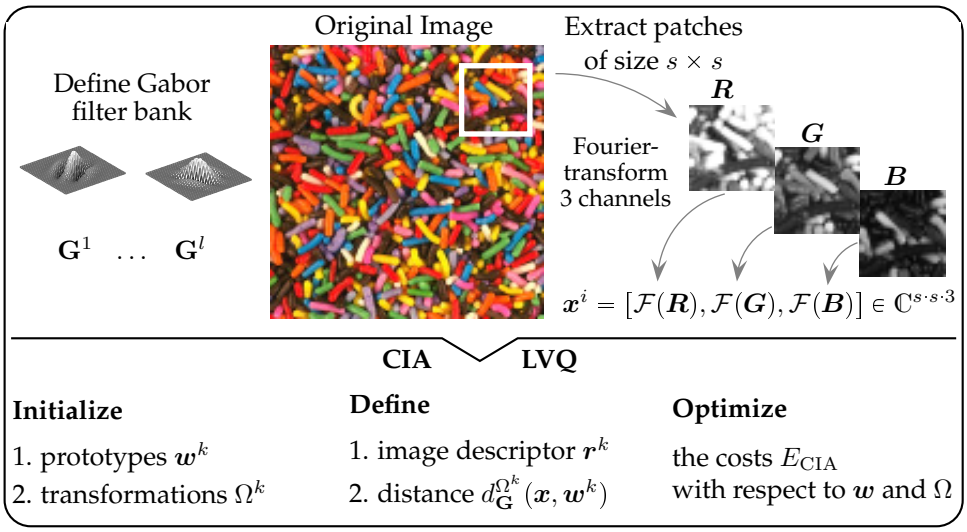


Figure 5.1: Methodology overview for the proposed CIA LVQ.

is often considered as a different class, here we distinguish into five different classes equivalent to the five aforementioned groups. Despite its increased difficulty, this classification task allows us to better demonstrate the ability of CIA LVQ to describe general characteristics of real-world texture patterns.

At first we draw 15×15 patches randomly from each image shown in Fig. 5.2. The training set contains 150 patches per image, resulting in 3000 samples in total, while the test set holds 50 patches from each image. The test set may contain patches

Figure 5.2: Images, which are used to provide random patches for training and test.

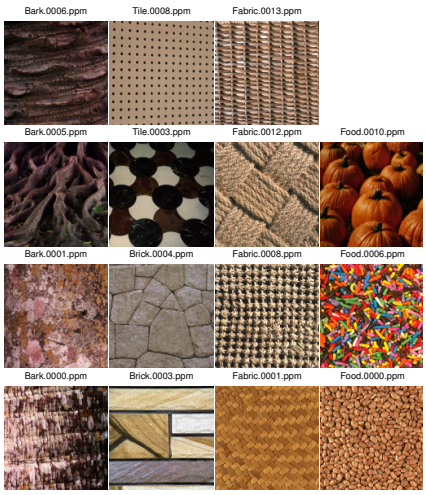


Figure 5.3: Images used to provide random patches for evaluation.

which partially overlap with those used for training. Therefore the images in Fig. 5.3 are used in order to create an evaluation set that was never seen in the training process. The evaluation set consists of 50 randomly drawn patches per image and is used to show the generalization ability of the approach.

A note is due here to the nature of the filter used. A 2D Gabor filter is defined as a Gaussian kernel function modulated by a sinusoidal plane wave. All filter kernels can be generated from one basic wavelet by dilation and rotation. In this experiment our filter bank consists of 12 Gabor filters of bandwidth equal to 1 at six orientations $\theta = 0$, 30, 60, 90, 120 and 150 degrees and two scales (wavelegths) varying by one octave: $\lambda = 7$ and $7\sqrt{2}$. These scales ensure that the Gabor function yields an adequate number of visible parallel excitatory and inhibitory stripe zones. Dependent on the patch size different scales might be adequate. We set the phase offset $\phi = 0$ and the aspect ratio $\gamma = 1$ for all filters. In this way we create center-on symmetric filters with circular support. We run the CIA LVQ with class-wise matrices Ω^c initialized with the identity matrix and 4 prototypes per class for $t_{\max}$ =300 epochs. The learning rates were chosen as

$$\tau_1(t) = 0.002\,(0.005)^{t/t_{\max}} \tag{5.4}$$

$$\tau_2(t) = 10^{-3}\left(10^{-2}\right)^{t/t_{\max}}\ , \tag{5.5}$$

where t is the current epoch. Using more filters and more localized matrices Ω^j may cause overfitting effects. So it is advisable to increase the complexity of the system

Table 5.1: Confusion matrices (eval. set)

CIA-LVQ:

	1	2	3	4	5	$\sum$
1	176	10	12	7	2	207
2	1	57	11	9	3	81
3	18	25	43	31	10	127
4	1	5	23	127	4	160
5	4	3	11	26	131	175
$\sum$	200	100	100	200	150	750

class-wise accuracy of estimation in %

88.00 57.00 43.00 63.50 87.33

RGB2G:

	1	2	3	4	5	$\sum$
1	52	14	7	36	28	137
2	51	45	30	37	34	197
3	27	27	51	26	22	153
4	29	6	8	83	11	137
5	41	8	4	18	55	126
$\sum$	200	100	100	200	150	750

class-wise accuracy of estimation in %

26.00 45.00 51.00 41.50 36.67

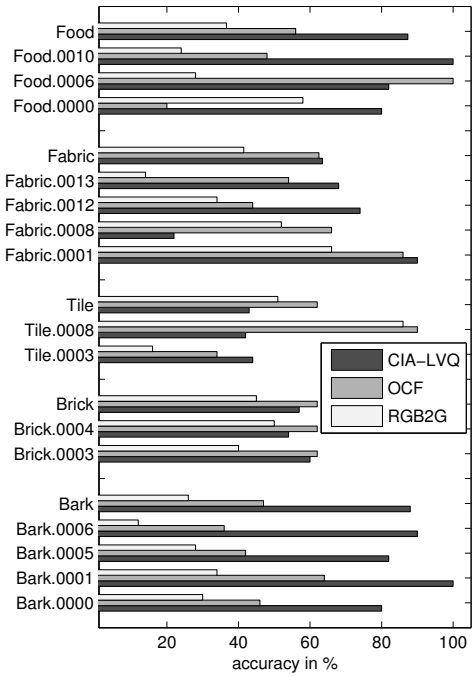


Figure 5.4: Class-wise and individual image accuracies

carefully. The training error is 10.6% and the error on the test set 28%.

We use the same data sets and the same filter bank to compare with the common approach of deriving textural information only from the luminance plane of images (Drimbarean and Whelan 2001). This approach is considered to often outperform combined color and texture features (Mäenpää and Pietikäinen 2004). For comparison, we also use an RGB to gray (RGB2G) transformation, which builds intensity values by a weighted sum of the color components of every pixel:

$$0.2989 \cdot R + 0.587 \cdot G + 0.114 \cdot B \ . \tag{5.6}$$

We vectorize all patches $\boldsymbol{x}$ and in this case the image patch descriptor is given by

$$\boldsymbol{r}_2(\boldsymbol{x}) = \sum_l \boldsymbol{x} * \mathbf{G}^l \ . \tag{5.7}$$

We use a Nearest Neighbor (1-NN) classification scheme with a dissimilarity mea-

sure similar to Eq. (5.2):

$$d_{\mathbf{G}}(\boldsymbol{x}^i, \boldsymbol{x}^j) = \| \, |\boldsymbol{r}_2(\boldsymbol{x}^i)|^2 - |\boldsymbol{r}_2(\boldsymbol{x}^j)|^2 \, \|^2 \ . \tag{5.8}$$

The 1-NN scheme based on the RGB2G transformation shows a test error of 37.5%, but most interesting is the comparison of the classification errors on the evaluation set. Here the 1-NN scheme shows an error of 61.9%, while the CIA-LVQ still has an error of 28.8%. The LVQ scheme displays very good generalization, which is shown in Table 5.1 and Fig. 5.4. Note, that the accuracy rates among individual images of the same class can vary. Brick and Tile are the most difficult classes, because the texture is large, so it cannot be captured very well with such a small patch size, since a lot of patches might be drawn from non-textured regions. On the other side, classes like Food and Bark with less diversity regarding textural structures can be learned quite well.

The prototypes, which classify the evaluation set are shown in Fig. 5.5. Additionally we show some example patches from the evaluation set, which are classified correctly together with their descriptors in Fig. 5.6 and some examples of wrongly classified patches in Fig. 5.7. Some obvious problems occur due to the random sampling and the very small patchsize: a lot of samples of Brick and Tile, for example, show homogeneous regions coming from the area in-between the textural structure (see Fig. 5.7). We observe, that classes which vary a lot in the size of the actual structure (e.g. Brick and Tile) are more difficult to recognize than classes with small variations in the scale of texture (like Bark and Food). It is interesting to notice that random patches drawn from Food.0010.ppm are 100% correctly classified, even though no patch from this image was ever used to train the algorithm. The learned local transformation recognizes the channels leading to the orange color and increased their weights to distinguish this class from others.

5.4 Conclusion and outlook

In this contribution we proposed a prototype based framework for color texture analysis. In contrary to standard approaches which are either based on a single channel representation of the images through a fixed transformation or empirical observations for combining color and textural information, we offer the alternative of data driven learning of suitable, parameterized image descriptors. The ability of weighting different color channels automatically according to their importance for the classification task is the most important factor which distinguishes our approach. We have formulated a novel general principle: based on a differentiable convolution and a predefined filter bank the CIA-LVQ algorithm optimizes the classification. It is also of conceptual value that this adaptation of LVQ is suitable for

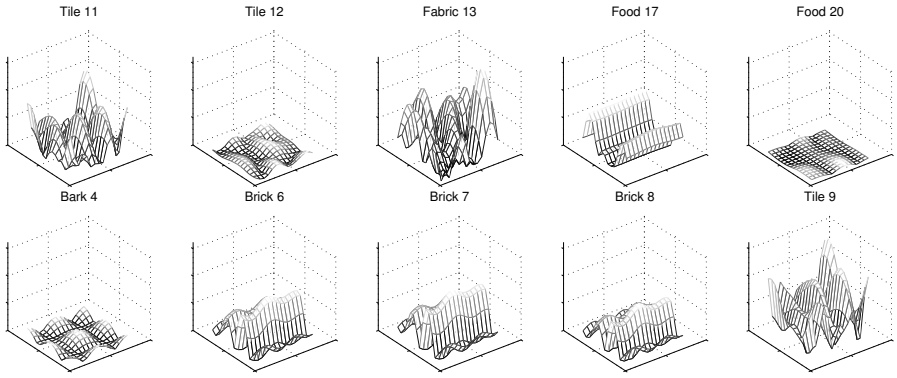


Figure 5.5: Magnitude of the descriptors $|\boldsymbol{r}_L(\boldsymbol{w}^L)|$ of the prototypes which classify the evaluation set.

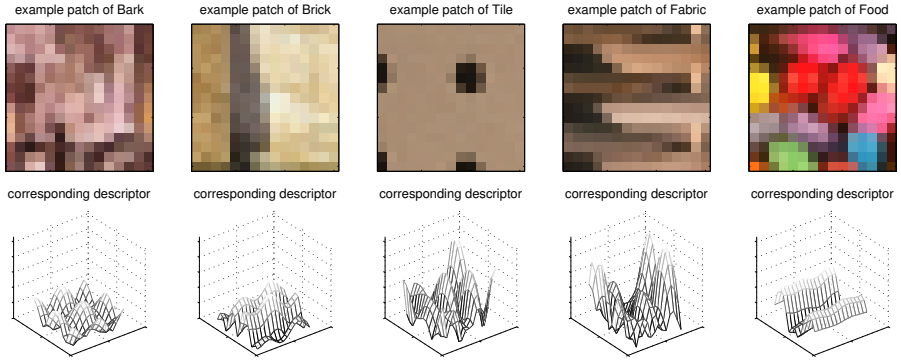


Figure 5.6: Magnitude of the descriptors $|\boldsymbol{r}^L(\boldsymbol{w}^L)|$ of some correct classified example patches of the evaluation set.

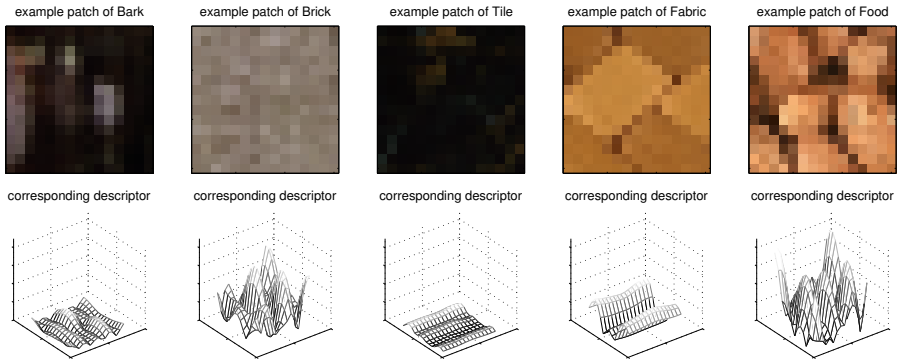


Figure 5.7: Magnitude of the descriptors $|\boldsymbol{r}^L(\boldsymbol{w}^L)|$ of some wrongly classified example patches of the evaluation set.

learning in the complex numbers domain. As an example we used Gabor filters to classify texture patterns in 15×15 patches randomly drawn from images of the VisTex database. The results show that the algorithm can learn typical texture patterns with very good generalization, even from relatively small patches and filter banks. Similarly to Gabor filters any other family of 2D filters commonly used to describe gray scale image information could be adapted and applied to color image analysis with this algorithm. A filter bank with differences of Gaussians for color edge detection is a possible example. Investigation of the performance of the system on other filters can be addressed in future. Furthermore, depending on the task it might be desirable that two patches in which the same texture occurs on different positions should not be interpreted as similar. In this case another similarity measure should be used: $\| \, |\boldsymbol{r}(\boldsymbol{x}^i) - \boldsymbol{r}(\boldsymbol{w}^L)| \, \|^2$, which is not based on the difference of magnitudes. This might be of advantage for example in the recognition of objects such as traffic signs, were a corner or an edge might have different meanings dependent on its position in the image.

Furthermore, the algorithm theoretically allows the optimization with respect to all variables. Using a dissimilarity measure

$$d_{\mathbf{F}}^{\Omega^k}(\boldsymbol{x}, \boldsymbol{w}^k) = \| \, |\Omega^k \boldsymbol{x} * \mathbf{F}^k|^2 \, - \boldsymbol{w}^k\|^2 \tag{5.9}$$

in the cost function E_{CIA} Eq. (5.3) and performing an optimization with respect to the prototypes $\boldsymbol{w}$, matrices Ω and the local filters $\mathbf{F}$ showed already promising results. Here, the matrix $\mathbf{F}$ can be initialized e.g. as the sum of differently parameterized Gabor filters. During the training unnecessary scales and orientations are suppressed, which yields individually suitable class-wise filter banks. The investigation of this extension will be addressed in forthcoming projects.

5.A Derivatives of CIA LVQ

Here we show the derivatives of the CIA LVQ costfunction E_{CIA}, see Eq. (5.3), for one presented training example $\boldsymbol{x}^i$, with respect to the prototypes $\boldsymbol{w}^L$ and the transformation matrices $\Omega^L \in \mathbb{R}^{M\times N}$ with $L \in \{J, K\}$. In the following we denote the real part of a variable $\boldsymbol{v}$ by $\Re(\boldsymbol{v})$ and the imaginary part by $\Im(\boldsymbol{v})$. We have to take the derivatives with respect to the real and imaginary part, respectively:

$$\frac{\partial E_{\mathrm{CIA}}}{\partial \boldsymbol{w}^L} = \frac{\partial E_{\mathrm{CIA}}}{\partial \Re(\boldsymbol{w}^L)} + i\frac{\partial E_{\mathrm{CIA}}}{\partial \Im(\boldsymbol{w}^L)} = \frac{\partial E_{\mathrm{CIA}}}{\partial d_{\mathbf{G}}^{\Omega^L}} \cdot \left(\frac{\partial d_{\mathbf{G}}^{\Omega^L}}{\partial \Re(\boldsymbol{w}^L)} + i\frac{\partial d_{\mathbf{G}}^{\Omega^L}}{\partial \Im(\boldsymbol{w}^L)} \right) \tag{5.10}$$

$$\frac{\partial E_{\mathrm{CIA}}}{\partial \Omega^L} = \frac{\partial E_{\mathrm{CIA}}}{\partial d_{\mathbf{G}}^{\Omega^L}} \cdot \left(\frac{\partial d_{\mathbf{G}}^{\Omega^L}}{\partial \Re(\Omega^L)} + i\frac{\partial d_{\mathbf{G}}^{\Omega^L}}{\partial \Im(\Omega^L)} \right) \tag{5.11}$$

$$\text{with} \quad \gamma_{\mathbf{G}}^J = \frac{\partial E_{\mathrm{CIA}}}{\partial d_{\mathbf{G}}^{\Omega^J}} = \frac{2 \cdot d_{\mathbf{G}}^{\Omega^K}(\boldsymbol{x}^i, \boldsymbol{w}^K)}{\left(d_{\mathbf{G}}^{\Omega^J}(\boldsymbol{x}^i, \boldsymbol{w}^J) + d_{\mathbf{G}}^{\Omega^K}(\boldsymbol{x}^i, \boldsymbol{w}^K)\right)^2} \tag{5.12}$$

$$\gamma_{\mathbf{G}}^K = \frac{\partial E_{\mathrm{CIA}}}{\partial d_{\mathbf{G}}^{\Omega^K}} = \frac{-2 \cdot d_{\mathbf{G}}^{\Omega^J}(\boldsymbol{x}^i, \boldsymbol{w}^J)}{\left(d_{\mathbf{G}}^{\Omega^J}(\boldsymbol{x}^i, \boldsymbol{w}^J) + d_{\mathbf{G}}^{\Omega^K}(\boldsymbol{x}^i, \boldsymbol{w}^K)\right)^2} \ . \tag{5.13}$$

The derivatives can be written as:

$$\frac{\partial E_{\mathrm{CIA}}}{\partial \boldsymbol{w}^L} = -4\gamma_{\mathbf{G}}^L \left[\left[|\boldsymbol{r}^L(\boldsymbol{x}^i)|^2 - |\boldsymbol{r}^L(\boldsymbol{w}^L)|^2 \right] \cdot \boldsymbol{r}^L(\boldsymbol{w}^L)^* * \left[\sum_l \Omega^L * \mathbf{G}^l \right] \right]^* \tag{5.14}$$

$$\begin{aligned} \frac{\partial E_{\mathrm{CIA}}}{\partial \Omega^L} = \gamma_{\mathbf{G}}^L \Bigg(& 4\left(|\boldsymbol{r}^L(\boldsymbol{x}^i)|^2 - |\boldsymbol{r}^L(\boldsymbol{w}^L)|^2 \right) \\ & \cdot \left[\boldsymbol{r}^L(\boldsymbol{x}^i)^* * \left(\sum_l \boldsymbol{x}^i * \mathbf{G}^l \right) - \boldsymbol{r}^L(\boldsymbol{w}^L)^* * \left(\sum_l \boldsymbol{w}^L * \mathbf{G}^l \right) \right] \Bigg)^* \ , \end{aligned} \tag{5.15}$$

with * denoting the complex conjugate. A more detailed description of the derivatives is achieved by rewriting the distance Eq. (5.2):

$$\begin{aligned} d_{\mathbf{G}}^{\Omega^L}(\boldsymbol{x}, \boldsymbol{w}^L) &= \| \Re(\boldsymbol{r}^L(\boldsymbol{x}))^2 + \Im(\boldsymbol{r}^L(\boldsymbol{x}))^2 - \Re(\boldsymbol{r}^L(\boldsymbol{w}^L))^2 - \Im(\boldsymbol{r}^L(\boldsymbol{w}^L))^2 \|^2 \\ &= \sum_{m=1}^{M} \left(\Re(\boldsymbol{r}_m^L(\boldsymbol{x}))^2 + \Im(\boldsymbol{r}_m^L(\boldsymbol{x}))^2 - \Re(\boldsymbol{r}_m^L(\boldsymbol{w}^L))^2 - \Im(\boldsymbol{r}_m^L(\boldsymbol{w}^L)) \right)^2 \end{aligned} \tag{5.16}$$

$$\begin{aligned} \Re(\boldsymbol{r}_m^L(\boldsymbol{v})) = & \sum_l \Re(\mathbf{G}_m^l) \left(\sum_j \Im(v_j) \cdot \Im(\Omega_{mj}^L) - \sum_j \Re(v_j) \cdot \Re(\Omega_{mj}^L) \right) \\ & + \sum_l \Im(\mathbf{G}_m^l) \left(\sum_j \Im(v_j) \cdot \Re(\Omega_{mj}^L) + \sum_j \Re(v_j) \cdot \Im(\Omega_{mj}^L) \right) \end{aligned} \tag{5.17}$$

$$\Im(\boldsymbol{r}_m^L(\boldsymbol{v})) = \sum_l \Re(\mathbf{G}_m^l) \left(-\sum_j \Re(v_j) \cdot \Im(\Omega_{mj}^L) - \sum_j \Im(v_j) \cdot \Re(\Omega_{mj}^L) \right) + \sum_l \Im(\mathbf{G}_m^l) \left(\sum_j \Im(v_j) \cdot \Im(\Omega_{mj}^L) - \sum_j \Re(v_j) \cdot \Re(\Omega_{mj}^L) \right) . \tag{5.18}$$

The derivatives with respect to the real and imaginary parts of one element of the prototypes $\boldsymbol{w}_r^L$ and matrices Ω_{mn}^L read:

$$\begin{aligned}
\frac{\partial d_{\mathbf{G}}^{\Omega^L}}{\partial \Re(w_r^L)} =& 2 \sum_{m=1}^{M} \left(\Re(\boldsymbol{r}_m^L(\boldsymbol{x}))^2 + \Im(\boldsymbol{r}_m^L(\boldsymbol{x}))^2 - \Re(\boldsymbol{r}_m^L(\boldsymbol{w}^L))^2 - \Im(\boldsymbol{r}_m^L(\boldsymbol{w}^L)) \right) \cdot \\
& \Bigg(-2 \cdot \Bigg[\sum_l \Re(\mathbf{G}_m^l) \left(\sum_j \Im(w_j^L) \cdot \Im(\Omega_{mj}^L) - \Re(w_j^L) \cdot \Re(\Omega_{mj}^L) \right) + \\
& \quad \sum_l \Im(\mathbf{G}_m^l) \left(\sum_j \Im(w_j^L) \cdot \Re(\Omega_{mj}^L) + \Re(w_j^L) \cdot \Im(\Omega_{mj}^L) \right) \Bigg] \cdot \\
& \quad \left(\sum_l -\Re(\mathbf{G}_m^l) \cdot \Re(\Omega_{mr}^L) + \Im(\mathbf{G}_m^l) \cdot \Im(\Omega_{mr}^L) \right) \\
& -2 \cdot \Bigg[\sum_l \Re(\mathbf{G}_m^l) \left(\sum_j -\Re(w_j^L) \cdot \Im(\Omega_{mj}^L) - \Im(w_j^L) \cdot \Re(\Omega_{mj}^L) \right) + \\
& \quad \sum_l \Im(\mathbf{G}_m^l) \left(\sum_j \Im(w_j^L) \cdot \Im(\Omega_{mj}^L) - \Re(w_j^L) \cdot \Re(\Omega_{mj}^L) \right) \Bigg] \cdot \\
& \quad \left(\sum_l -\Re(\mathbf{G}_m^l) \cdot \Im(\Omega_{mr}^L) - \Im(\mathbf{G}_m^l) \cdot \Re(\Omega_{mr}^L) \right) \Bigg) \\
\frac{\partial d_{\mathbf{G}}^{\Omega^L}}{\partial \Im(w_r^L)} =& 2 \sum_{m=1}^{M} \left(\Re(\boldsymbol{r}_m^L(\boldsymbol{x}))^2 + \Im(\boldsymbol{r}_m^L(\boldsymbol{x}))^2 - \Re(\boldsymbol{r}_m^L(\boldsymbol{w}^L))^2 - \Im(\boldsymbol{r}_m^L(\boldsymbol{w}^L)) \right) \cdot \\
& \Bigg(-2 \cdot \Bigg[\sum_l \Re(\mathbf{G}_m^l) \left(\sum_j \Im(w_j^L) \cdot \Im(\Omega_{mj}^L) - \Re(w_j^L) \cdot \Re(\Omega_{mj}^L) \right) + \\
& \quad \sum_l \Im(\mathbf{G}_m^l) \left(\sum_j \Im(w_j^L) \cdot \Re(\Omega_{mj}^L) + \Re(w_j^L) \cdot \Im(\Omega_{mj}^L) \right) \Bigg] \cdot \\
& \quad \left(\sum_l \Re(\mathbf{G}_m^l) \cdot \Im(\Omega_{mr}^L) + \Im(\mathbf{G}_m^l) \cdot \Re(\Omega_{mr}^L) \right) \\
& -2 \cdot \Bigg[\sum_l \Re(\mathbf{G}_m^l) \left(\sum_j -\Re(w_j^L) \cdot \Im(\Omega_{mj}^L) - \Im(w_j^L) \cdot \Re(\Omega_{mj}^L) \right) + \quad \rightsquigarrow
\end{aligned}$$

$$\rightsquigarrow \quad \sum_l \Im(\mathbf{G}_m^l)\left(\sum_j \Im(w_j^L)\cdot\Im(\Omega_{mj}^L) - \Re(w_j^L)\cdot\Re(\Omega_{mj}^L)\right)\Bigg]\cdot$$
$$\left(\sum_l -\Re(\mathbf{G}_m^l)\cdot\Re(\Omega_{mr}^L) + \Im(\mathbf{G}_m^l)\cdot\Im(\Omega_{mr}^L)\right)\Bigg) \quad .$$

$$\frac{\partial d_{\mathbf{G}}^{\Omega^L}}{\partial \Re(\Omega_{mn}^L)} = 2\left(\Re(\boldsymbol{r}_m^L(\boldsymbol{x}))^2 + \Im(\boldsymbol{r}_m^L(\boldsymbol{x}))^2 - \Re(\boldsymbol{r}_m^L(\boldsymbol{w}^L))^2 - \Im(\boldsymbol{r}_m^L(\boldsymbol{w}^L))\right)\cdot$$
$$\Bigg(2\cdot\Bigg[\sum_l \Re(\mathbf{G}_m^l)\left(\sum_j \Im(x_j)\cdot\Im(\Omega_{mj}^L) - \Re(x_j)\cdot\Re(\Omega_{mj}^L)\right) +$$
$$\sum_l \Im(\mathbf{G}_m^l)\left(\sum_j \Im(x_j)\cdot\Re(\Omega_{mj}^L) + \Re(x_j)\cdot\Im(\Omega_{mj}^L)\right)\Bigg]\cdot$$
$$\left(\sum_l -\Re(\mathbf{G}_m^l)\cdot\Re(x_n) + \Im(\mathbf{G}_m^l)\cdot\Im(x_n)\right) +$$
$$2\cdot\Bigg[\sum_l \Re(\mathbf{G}_m^l)\left(\sum_j -\Re(x_j)\cdot\Im(\Omega_{mj}^L) - \Im(x_j)\cdot\Re(\Omega_{mj}^L)\right) +$$
$$\sum_l \Im(\mathbf{G}_m^l)\left(\sum_j \Im(x_j)\cdot\Im(\Omega_{mj}^L) - \Re(x_j)\cdot\Re(\Omega_{mj}^L)\right)\Bigg]\cdot$$
$$\left(\sum_l -\Re(\mathbf{G}_m^l)\cdot\Im(x_n) - \Im(\mathbf{G}_m^l)\cdot\Re(x_n)\right)$$
$$-2\cdot\Bigg[\sum_l \Re(\mathbf{G}_m^l)\left(\sum_j \Im(w_j^L)\cdot\Im(\Omega_{mj}^L) - \Re(w_j^L)\cdot\Re(\Omega_{mj}^L)\right) +$$
$$\sum_l \Im(\mathbf{G}_m^l)\left(\sum_j \Im(w_j^L)\cdot\Re(\Omega_{mj}^L) + \Re(w_j^L)\cdot\Im(\Omega_{mj}^L)\right)\Bigg]\cdot$$
$$\left(\sum_l -\Re(\mathbf{G}_m^l)\cdot\Re(w_n^L) + \Im(\mathbf{G}_m^l)\cdot\Im(w_n^L)\right)$$
$$-2\cdot\Bigg[\sum_l \Re(\mathbf{G}_m^l)\left(\sum_j -\Re(x_j)\cdot\Im(\Omega_{mj}^L) - \Im(x_j)\cdot\Re(\Omega_{mj}^L)\right) +$$
$$\sum_l \Im(\mathbf{G}_m^l)\left(\sum_j \Im(x_j)\cdot\Im(\Omega_{mj}^L) - \Re(x_j)\cdot\Re(\Omega_{mj}^L)\right)\Bigg]\cdot$$
$$\left(\sum_l -\Re(\mathbf{G}_m^l)\cdot\Im(w_n^L) - \Im(\mathbf{G}_m^l)\cdot\Re(w_n^L)\right)\Bigg) \quad .$$

Part II

Dimension Reduction and Visualization

Published as:

K. Bunte, M. Biehl and B. Hammer – *"A general framework for dimensionality reducing data visualization using explicit mapping functions,"* accepted for publication in Neural Computation 2011.

K. Bunte, M. Biehl and B. Hammer – *"Supervised dimension reduction mappings,"* in Proc. of European Symposium on Artificial Neural Networks (ESANN), pp. 281–286, Bruges, Belgium, April 2011.

K. Bunte, M. Biehl and B. Hammer – *"Dimensionality reduction mappings,"* in IEEE Symposium on Computational Intelligence and Data Mining (CIDM), pp. 349–356, Paris, France, April 2011.

Chapter 6

Dimension Reduction Mappings

Any intelligent fool can make things bigger and more complex . . . It takes a touch of genius - and a lot of courage to move in the opposite direction.

Albert Einstein

Abstract

In recent years a wealth of dimension reduction techniques for data visualization and preprocessing has been established. Non-parametric methods require additional effort for out-of-sample extensions, because they just provide a mapping of a given finite set of points. In this chapter we propose a general view on non-parametric dimension reduction based on the concept of cost functions and properties of the data. Based on this general principle we transfer non-parametric dimension reduction to explicit mappings of the data manifold such that direct out-of-sample extensions become possible. Furthermore, this concept offers the possibility to investigate the generalization ability of data visualization to new data points. We demonstrate the approach based on a simple global linear mapping as well as prototype-based local linear mappings. In addition, we can bias the functional form according to given auxiliary information. This leads to explicit supervised visualization mappings which discriminative properties are comparable to state-of-the-art approaches.

6.1 Introduction

Due to improved sensor technology, dedicated data formats and rapidly increasing digitalization capabilities the amount of electronic data increases dramatically since decades (Frawley et al. 1991). As a consequence, manual inspection of

digital data sets often becomes infeasible. Automatic methods which help users to quickly scan through large amounts of data are desirable. In recent years, many powerful non-linear dimension reduction techniques have been developed which provide a visualization of complex data sets. This way, humans can rely on their astonishing cognitive capabilities for visual perception when extracting information from large data volumes: structural characteristics can be captured almost instantly by humans independent of the number of displayed data points.

In the past years many powerful dimension reduction techniques have been proposed (Lee and Verleysen 2007, van der Maaten et al. 2009, van der Maaten and Hinton 2008, Venna et al. 2010). Basically, the task of dimensionality reduction is to represent data points contained in a high-dimensional data manifold by low-dimensional counterparts in two or three dimensions, while preserving as much information as possible. Since it is not clear, a priori, which parts of the data are relevant to the user, this problem is inherently ill-posed: depending on the specific data domain and the situation at hand, different aspects can be the focus of attention. Therefore a variety of different methods has been proposed which try to preserve different properties of the data and which impose additional regularizing constraints on the methods: Spectral techniques such as Locally Linear Embedding (LLE) (Roweis and Saul 2000), Isomap (Tenenbaum et al. 2000), or Laplacian Eigenmaps (Belkin and Niyogi. 2003) rely on the spectrum of the neighborhood graph of the data and preserve important properties of this graph. In general a unique algebraic solution of the corresponding mathematical objective can be formalized. To arrive at unimodal costs, these methods often base on very simple affinity functions such as Gaussians. As a consequence their results can be flawed when it comes to boundaries, disconnected manifolds, or holes. Using more complex affinities such as geodesic distance or local neighborhood relations, techniques such as Isomap or Maximum Variance Unfolding (MVU) (Weinberger and Saul 2006) can partially avoid these problems at the prize of higher computational costs. Many highly non-linear techniques have been proposed as an alternative which often suffer from the existence of local minima. They do not yield unique solutions, and they require numerical optimization techniques. In turn, due to the greater complexity, their visualization properties may be superior as demonstrated in (Hinton and Roweis 2003, van der Maaten and Hinton 2008, Carreira-Perpiñán 2010).

All methods mentioned above map a given finite set of data points to low-dimensions. Additional effort is required to include new points into the mapping and to arrive at out-of-sample extensions: usually, novel points are mapped to the projection space by minimizing the underlying cost function of the visualization method while keeping the projections of the priorly given data points fixed. This way novel coordinates depend on all given data points, and the effort to map new

data depends on the size of the training set. Moreover, no explicit mapping function is available and the generalization ability of the techniques to novel data is not clear.

As an alternative, some approaches derive an explicit function that maps the given data to low-dimension. This way, an immediate extension to novel data becomes possible. Linear techniques such as standard Principal Component Analysis (PCA) or Fisher Discriminant Analysis (FDA) provide an explicit mapping. Autoencoder networks can be seen as a non-linear extension of PCA which directly aims at the inference of a non-linear mapping function and its approximate inverse. Nonlinear mapping functions have also been considered by (Bae et al. 2010) where only few points are mapped using a dimensionality reduction technique and an interpolation to all data is done by means of a k-NN approach. For LLE, a similar extension has been proposed based on locally linear functions by (Roweis and Saul 2000) called Locally Linear Coordination (LLC). There, the function parameters are optimized directly using the LLE cost function. Similarly, t-distributed SNE (t-SNE) has been extended to a mapping given by deep encoder networks (van der Maaten 2009), relying on the t-SNE cost function to optimize the mapping function parameters. In (Suykens 2008) a kernel mappings with a reference point is used to arrive at high-quality data visualization mappings. They also experimentally demonstrate the excellent generalization ability and visualization properties of the technique. Albeit these approaches constitute promising directions to arrive at explicit dimensionality reduction mappings, many of the techniques have been developed for a specific setting and dimensionality reduction technique only.

In this chapter we propose a general principle to formalize non-parametric dimension reduction based on cost optimization. This general principle allows us to simultaneously extend non-parametric methods to explicit mapping functions for which out-of-sample extensions are immediate. In this setting, the functional form of the mapping is fixed a priori and function parameters are optimized within the dimension reduction framework instead of the coordinates of single point projections. We demonstrate the suitability of this approach using two different types of functions: simple linear projections and locally linear functions. Interestingly, it can be shown that state of the art dimensionality reduction cost functions as provided by t-SNE, for example, can even improve simple linear dimensionality reduction functions as compared to classical PCA. Furthermore, the performance of state-of-the-art techniques such as presented by (van der Maaten 2009) can be achieved using more complex locally linear functions. Several benefits arise from an explicit dimension reduction mapping: out-of-sample extensions are immediate and require only constant time depending on the chosen form of the mapping. Since an explicit mapping function is available, approximate inverse mapping is possible at least locally: locally linear functions, for example, can be inverted using the pseudo-inverse. This

makes a deeper investigation of the structure of the projection possible. Depending on the form of the mapping function, only few parameters need to be determined and implicit regularization takes place. In consequence, only few data points are necessary to adequately determine these mapping parameters and generalize to novel data points. Hence, only a small subset of the full data is necessary for training, an enormous speed-up for large data sets is possible: Instead of a, usually, quadratic complexity to map the data, due to the computation of the pairwise distances, the mapping function can be determined in constant time complexity. The full data set can be displayed in linear time complexity. This opens the way to feasible dimension reduction for very large data sets. In this contribution, we experimentally demonstrate the suitability of the approach and we investigate the generalization ability in terms of several application. Moreover, we substantiate the experimental findings with an explicit mathematical formalization of the generalization ability of dimensionality reduction in the framework of statistical learning theory. Albeit we are not yet able to provide good explicit generalization bounds, we argue that principled learnability can be guaranteed for standard techniques. Another benefit of an explicit mapping function is the possibility to bias the dimensionality reduction mapping according to given prior knowledge. The task of dimension reduction is inherently ill-posed, and which aspects of the data are relevant for the user depends on the situation at hand. One way to shape the ill-posed task of data visualization is by incorporating auxiliary information as proposed e.g. by (Kaski et al. 2001).

There exist a few classical dimension reducing visualization tools which take class labeling into account: Feature selection can be interpreted as a particularly simple form of discriminative dimensionality reduction, see e.g. (Guyon and Elisseeff 2003) for an overview. Classical LDA as well as partial Least Squares regression (PLS) offer supervised linear visualization techniques based on the covariances of the classes; kernel techniques extend these settings to non-linear projections (Ma et al. 2007, Baudat and Anouar 2000). The principle of adaptive metrics used for data projection according to the given auxiliary information has been introduced in (Kaski et al. 2001, Peltonen et al. 2004). The obtained metric can be integrated into diverse techniques such as Self-organizing Map (SOM), Multidimensional Scaling (MDS), or a recent information theoretic model for data visualization (Kaski et al. 2001, Peltonen et al. 2004, Venna et al. 2010). An ad hoc metric adaptation is used in (Geng et al. 2005) to extend Isomap to class labels. Furthermore, in Chapter 7 of this thesis we discuss the combination of some metric adaptation schemes introduced in Part I with several examples of dimension reduction techniques (Bunte, Hammer, Wismüller and Biehl 2010). Alternative approaches change the cost function of dimensionality reduction, see (Iwata et al. 2007, Memisevic and Hinton 2005, Song et al. 2008) for examples. In this Chapter, we will show that auxiliary information in

the form of given class labels can be easily integrated into the dimension reduction scheme by biasing the functional form accordingly. As a result, one obtains a discriminative dimensionality reduction technique which is competitive to alternative state-of-the-art approaches.

We first shortly review several popular non-parametric dimensionality reduction techniques. We put them into a general framework based on the notion of cost functions which compare characteristics of the data and the projections. This general framework allows us to simultaneously extend the methods to explicit mappings which do not only lead to a finite set of projection coordinates but employ to an explicit projection function. We demonstrate this principle using a linear mapping and locally linear projections the form of which are induced by standard clustering techniques. We incorporate these functional forms into the cost function of t-SNE. Interestingly, the results are superior compared to standard linear techniques such as PCA and alternative mapping functions as presented, e.g., by (van der Maaten 2009). Furthermore, we demonstrate that the functional form can be biased towards auxiliary label information by choosing the functional form on top of supervised classification. Finally, we argue that, based on the notion of a mapping function, generalization properties of dimension reduction can be formalized in the framework of computational learning theory.

6.2 Dimension reduction as cost optimization

In this section we shortly review some popular dimension reduction methods proposed in the literature. We assume high-dimensional data points are given: $\boldsymbol{x}^i \in \mathbb{R}^N$ where $i = 1 \ldots n$. These points are projected to a low-dimensional embedding space $\mathcal{E} \in \mathbb{R}^M$, with $M < N$, usually $M \in \{2, 3\}$ for visualization. The coordinates of the points in the projection space are referred to as $\boldsymbol{\xi}^i \in \mathbb{R}^M$ for $i = 1, \ldots, n$. Furthermore, $\boldsymbol{\Xi}$ refers to the matrix of all points $\{\boldsymbol{\xi}^i\}_{i=1}^n$. Often, visualization techniques refer to the distances or affinities of data in the high-dimensional input space $\mathcal{X}$ and the projection space $\mathcal{E}$, respectively. The pairwise affinities are denoted as $d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{x}^j)$ for the original high-dimensional data points and by $d_{\mathcal{E}}(\boldsymbol{\xi}^i, \boldsymbol{\xi}^j)$ for the corresponding dissimilarities in the embedding space. Usually, $d_{\mathcal{E}}$ is chosen as Euclidean distance, while $d_{\mathcal{X}}$ is chosen according to the data set at hand, e.g. it is given by the Euclidean or the geodesic distance in the high-dimensional space. A mathematical formalization of dimensionality reduction can take place in different ways:

Multidimensional Scaling and Extensions:

MDS (Torgerson 1952) is probably one of the oldest dimension reduction methods. It aims at the preservation of pairwise relations measured in the least square sense.

The original MDS measures the pairwise relations of the data in terms of dot products in the original and the embedding space respectively and minimizes the cost function:

$$E_{\mathrm{MDS}} = \sum_{ij} ((\boldsymbol{x}^i)^\top \boldsymbol{x}^j - (\boldsymbol{\xi}^i)^\top \boldsymbol{\xi}^j)^2 \ . \tag{6.1}$$

The advantage of this formulation is that an analytical solution is available. In later approaches, the objective has been changed to the preservation of distances, often called goodness-of-fit or stress measure:

$$E_{\mathrm{MDS}} = \frac{1}{a} \sum_{ij} w_{ij} (d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{x}^j) - d_{\mathcal{E}}(\boldsymbol{\xi}^i, \boldsymbol{\xi}^j))^2 \tag{6.2}$$

with Euclidean distances $d_{\mathcal{X}}$ and $d_{\mathcal{E}}$ and a normalizing constant a (Lee and Verleysen 2007). The weights can be chosen for example as $w_{ij} = 1$. In the well-known Sammon mapping (Sammon 1969) they take the form $w_{ij} = 1/d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{x}^j)$, this way emphasizing the preservation of small distances. There, the constant a is set to the sum of the distances and the optimization takes place by a gradient descent procedure.

Isomap:

Depending on the actual data, the Euclidean distance might not be appropriate to describe pairwise relations. Therefore, Isomap (Geng et al. 2005) is based on the approximation of geodesic distances, which measure the relations along the data manifold. A neighborhood graph is constructed using k neighborhoods or ϵ-balls and the shortest path lengths in this graph (computed using Dijkstra's algorithm, for example) define the pairwise affinities $d_{\mathcal{X}}$ in the data space. Afterwards, the standard MDS procedure is used, which is mentioned above.

Locally Linear Embedding:

LLE (Roweis and Saul 2000) aims at the preservation of local topologies defined by the reconstruction of data points i by means of linear combination of its neighbors j. We denote the property that j is neighbor to i by $i \to j$. As for Isomap, local neighbors can be defined based on k-NNs or ϵ-balls, respectively. To obtain weights for reconstruction, the objective $\sum_i \left(\boldsymbol{x}^i - \sum_{j:\, i \to j} w_{ij} \boldsymbol{x}^j \right)^2$ in original space is minimized under the constraint $\sum_j w_{ij} = 1$, in oder to ensure rotation and translation invariance of the output. Afterwards, the projections are determined such that local linear relationships are preserved as well as possible in a least squared sense: minimize $\sum_i (\boldsymbol{\xi}^i - \sum_{j:\, i \to j} w_{ij} \boldsymbol{\xi}^j)^2$ subject to the constraints of centered coordinates

$\sum_i \boldsymbol{\xi}^i = 0$ with unit covariance $\boldsymbol{\Xi}^\top \boldsymbol{\Xi} = \mathbf{I}$. Where $\mathbf{I}$ is the $M \times M$ identity matrix. Here, the normalization of the reconstruction weights leads to a unique optimum of the system. The LLE method is summarized in Algorithm 6.1:

Algorithm 6.1 : Optimization problem for Locally Linear Embedding

Step 1: select neighbors $i \to j$
Step 2: obtain reconstruction weights
Minimize $\sum_i \left(\boldsymbol{x}^i - \sum_{j:\ i \to j} w_{ij} \boldsymbol{x}^j \right)^2$ **subject to:**
rotation and translation invariance: $\sum_j w_{ij} = 1$
Step 3: determine projections
Minimize $\sum_i (\boldsymbol{\xi}^i - \sum_{j:\ i \to j} w_{ij} \boldsymbol{\xi}^j)^2$ **subject to:**
(a) centered coordinates: $\sum_i \boldsymbol{\xi}^i = 0$
(b) unit covariance $\boldsymbol{\Xi}^\top \boldsymbol{\Xi} = \mathbf{I}$

Laplacian Eigenmaps:

Similar to LLE and Isomap, Laplacian Eigenmaps (Belkin and Niyogi. 2003) are based on the construction of a local neighborhood graph given the k-NNs or an ϵ-neighborhood, respectively. The connections are weighted by coefficients w_{ij}, e.g. using a heat kernel. The projection is obtained by solving a generalized eigenvalue problem given the corresponding graph Laplacian $L = \mathbf{A} - \mathbf{D}$ with adjacency matrix $\mathbf{A}$ and the degree matrix $\mathbf{D}$ of the graph, picking the eigendirections corresponding to the smallest eigenvalues unequal to 0. This is equivalent to minimizing the embedding objective

$$\sum_{i \to j} w_{ij} \cdot d_{\mathcal{E}}(\boldsymbol{\xi}^i, \boldsymbol{\xi}^j)^2 = 2 \boldsymbol{\Xi}^\top L\, \boldsymbol{\Xi} \quad \text{(considering Euclidean distance } d_{\mathcal{E}}) \tag{6.3}$$

under constraints $\boldsymbol{\Xi}^\top \mathbf{D}\, \boldsymbol{\Xi} = \mathbf{I}$ and $\boldsymbol{\Xi}^\top \mathbf{D}\, \mathbf{1} = \mathbf{0}$, where $\mathbf{D}$ is the degree matrix to remove scaling and translation factors. This objective is summarized in Algorithm 6.2:

Algorithm 6.2 : Optimization problem for Laplacian Eigenmaps

Construct neighborhood graph weighting the edges by w_{ij}
and determine graph Laplacian $L = \mathbf{A} - \mathbf{D}$
Minimize $\boldsymbol{\Xi}^\top L\, \boldsymbol{\Xi}$ **subject to:**
(a) $\boldsymbol{\Xi}^\top \mathbf{D}\, \boldsymbol{\Xi} = \mathbf{I}$
(b) $\boldsymbol{\Xi}^\top \mathbf{D}\, \mathbf{1} = \mathbf{0}$

Maximum Variance Unfolding:

MVU (Weinberger and Saul 2006) is based on a neighborhood graph with k nearest neighborhood graphs or ϵ-neighborhoods $\mathcal{N}$. Projections $\boldsymbol{\xi}^i$ are determined by maximizing the variance of the projection. The aim is, that neighboring points $\boldsymbol{x}^i$ and $\boldsymbol{x}^j$ preserve their affinities also in the low-dimensional space after projection: $d_{\mathcal{E}}(\boldsymbol{\xi}^i, \boldsymbol{\xi}^j) = d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{x}^j)$. Considering the inner product matrix $\mathbf{K} = (\boldsymbol{\Xi}^\top)\boldsymbol{\Xi}$ a reformulation as a convex problem is possible and a solution can be found in terms of a semidefinite program (SDP) (Vandenberghe and Boyd 1994). The variance is maximized by maximizing the trace of $\mathbf{K}$ (maximum variance unfolding) under constraints as summarized in Algorithm 6.3.

Algorithm 6.3 : Optimization problem for Maximum Variance Unfolding

Maximize $\max_{\mathbf{K}\geq 0} \operatorname{tr}(\mathbf{K})$ with $\mathbf{K} \in \mathbb{R}^{M\times M}$ **subject to:**

(a) preservation of distances:
$d_{\mathcal{E}}(\boldsymbol{\xi}^i, \boldsymbol{\xi}^j) = \mathbf{K}_{ii} + \mathbf{K}_{jj} - 2\mathbf{K}_{ij} = d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{x}^j) \quad \forall (i,j) \in \mathcal{N}$

(b) centered embedding data:
$\mathbf{K}\mathbf{1} = \mathbf{0}$, where $\mathbf{1} = (1, \ldots, 1)^\top$ and $\mathbf{0} = (0, \ldots, 0)^\top$

(c) $\mathbf{K} \succeq 0$

Numerous variants of the original formulation exist, where for example the distances are only allow to shrink or low-rank expansions of $\mathbf{K}$ are used to cope with the computational complexity of semidefinite programming. Furthermore, if a preservation of neighbored distances is not exactly possible, slack variables can be introduced.

Stochastic Neighbor Embedding:

Stochastic Neighbor Embedding (SNE) (Hinton and Roweis 2003) defines the characteristics of the data in terms of probabilities that i would pick j as neighbor in the original and embedding space respectively:

$$p_{j|i} = \frac{\exp\left(\frac{-d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{x}^j)^2}{2\sigma_i}\right)}{\sum_{k\neq i} \exp\left(\frac{-d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{x}^k)^2}{2\sigma_i}\right)} \tag{6.4}$$

$$\text{and} \quad q_{j|i} = \frac{\exp\left(-d_{\mathcal{E}}(\boldsymbol{\xi}^i, \boldsymbol{\xi}^j\right)^2}{\sum_{k\neq i} \exp\left(-d_{\mathcal{E}}(\boldsymbol{\xi}^i, \boldsymbol{\xi}^k)^2\right)} \tag{6.5}$$

using Euclidean distances as default. The objective

$$E_{\mathrm{SNE}} = -\sum_{ij} p_{j|i} \log \frac{p_{j|i}}{q_{j|i}} \tag{6.6}$$

corresponds to the Kullback-Leibler divergence between the probability densities in the original and the projection space. The bandwidths σ_i is either set by hand or is found by a binary search, such that the entropy of the distribution over neighbors becomes equal to $\log k$. Here k corresponds to the effective number of local neighbors, which is chosen by hand and in the following referred to as "perplexity". A gradient descent procedure is used for optimization, based on the derivative:

$$\frac{\partial E_{\mathrm{SNE}}}{\partial \boldsymbol{\xi}^i} = 2\sum_j (\boldsymbol{\xi}^i - \boldsymbol{\xi}^j)(p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j}) \ . \tag{6.7}$$

This can be interpreted as a sum of forces pulling $\boldsymbol{\xi}^i$ toward $\boldsymbol{\xi}^j$ or pushing it away depending on whether j is observed to be a neighbor more or less often than desired.

Algorithm 6.4 : Stochastic Neighbor Embedding (SNE)

1: determine σ_i (e.g. based on the perplexity) and compute probabilities $p_{j|i}$ (6.4)
2: initialize low dimensional images $\boldsymbol{\xi}$
3: **while** stopping criterion not reached **do**
4: compute probabilities $q_{j|i}$ Eq. (6.5)
5: update the $\boldsymbol{\xi}^i$ according to $\frac{\partial E_{\mathrm{SNE}}}{\partial \boldsymbol{\xi}^i}$ Eq. (6.7)
6: **end while**

T-Distributed Stochastic Neighbor Embedding:

t-SNE (van der Maaten and Hinton 2008) modifies the SNE cost function such that the long tailed student-t distribution is used in the embedding space instead of Gaussians. The cost function

$$E_{\mathrm{t-SNE}} = \sum_i \sum_j p_{ij} \log \left(\frac{p_{ij}}{q_{ij}} \right) \tag{6.8}$$

uses symmetrized conditional probabilities

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{6.9}$$

and

$$q_{ij} = \frac{(1 + d_{\mathcal{E}}(\boldsymbol{\xi}^i, \boldsymbol{\xi}^j)/\varsigma)^{-\frac{\varsigma+1}{2}}}{\sum_{k \neq l} (1 + d_{\mathcal{E}}(\boldsymbol{\xi}^k, \boldsymbol{\xi}^l)/\varsigma)^{-\frac{\varsigma+1}{2}}} \tag{6.10}$$

with n denoting the number of data points and the student-t distribution parameterized with $\varsigma = -1$ by default. Again, optimization is done in terms of a gradient method.

Algorithm 6.5 : t-distributed SNE (t-SNE)

same as for SNE using the student-t distribution in the embedding space and replacing the probabilities by Eqs. (6.9) and (6.10)

Neighborhood Retrieval Visualizer:

In (Venna et al. 2010) a quality measure for dimension reduction is derived from an information retrieval point of view is proposed. A new dimension reduction technique based on the new objective accompanies this proposal: the Neighborhood Retrieval Visualizer (NeRV). The cost function reads:

$$E_{\mathrm{NeRV}} = -c\sum_{ij} p_{j|i} \log \frac{p_{j|i}}{q_{j|i}} - (1-c)\sum_{ij} q_{j|i} \log \frac{q_{j|i}}{p_{j|i}} \tag{6.11}$$

with probabilities as defined for SNE (Eqs. (6.4) and (6.5)) and a weighting parameter $c \in [0,1]$ to control the influence of the competing terms related to the traditional measures precision and recall. The t-distributed NeRV (t-NeRV) extension is straightforward considering symmetric pairwise probabilities just as in t-SNE (Eqs. (6.9) and (6.10)) in the symmetrized version of the Kullback-Leibler divergence.

6.2.1 A general view

All methods as summarized above obey one general principle. Assume a finite sample of points $\mathbf{X} = (\boldsymbol{x}^i \in \mathbb{R}^N \mid i = 1,\ldots,n) = (\boldsymbol{x}^1,\ldots,\boldsymbol{x}^n)$ is given. These points should be mapped to a low-dimensional embedding space $\mathbb{R}^M$ with $M < N$, where data point $\boldsymbol{x}^i$ is mapped to the projection $\boldsymbol{\xi}^i \in \mathbb{R}^M$ by means of a non-parametric mapping. The projections are referred to as $\boldsymbol{\Xi} = (\boldsymbol{\xi}^i \mid i = 1,\ldots,n) = (\boldsymbol{\xi}^1,\ldots,\boldsymbol{\xi}^n)$. The sequence of tuples of data points and their projections is referred to as $\mathbf{X}\boldsymbol{\Xi} = ((\boldsymbol{x}^1,\boldsymbol{\xi}^1),\ldots,(\boldsymbol{x}^n,\boldsymbol{\xi}^n))$. We denote the set of all finite subsequences of $\mathbb{R}^N$ by $S(\mathbb{R}^N)$; more generally $S(A)$ refers to all finite subsequences of a given set A. Given a sequence $\mathbf{X} = (\boldsymbol{x}^1,\ldots,\boldsymbol{x}^n)$, its length is denoted by $n = |\mathbf{X}|$.

For all methods, the coefficients $\boldsymbol{\xi}^i$ are determined based on the same general principle, using the same basic ingredients, the characteristics derived from the original training set $\mathbf{X}$ for every data point, corresponding characteristics of its projection, and an error measure between these two characteristics. The latter is min-

imized during projection, possibly taking into account further constraints. More precisely, dimensionality reduction is characterized by the following ingredients:

- A function $\mathrm{char}_{\mathcal{X}} : S(\mathbb{R}^N) \times \mathbb{R}^N \to S(\mathbb{R})$ is fixed which maps a data sequence $\mathbf{X}$ and a point $\boldsymbol{x}$ in the original space $\mathbb{R}^N$ to a characteristic. Usually, $|\mathrm{char}_{\mathcal{X}}(\mathbf{X}, \boldsymbol{x})| = |\mathbf{X}|$.

- A function $\mathrm{char}_{\mathcal{E}} : S(\mathbb{R}^M \times \mathbb{R}^N) \times (\mathbb{R}^M \times \mathbb{R}^N) \to S(\mathbb{R})$ is fixed which maps a finite subset $\mathbf{X\Xi}$ of points and their projections, and a given tuple of a point and its projection to a corresponding characteristic.
  Usually $|\mathrm{char}_{\mathcal{E}}(\mathbf{X\Xi}, (\boldsymbol{x}^i, \boldsymbol{\xi}^i))| = |\mathbf{X\Xi}|$.

- An error measure is fixed which measures the difference of two such characteristics: $\mathrm{error} : S(\mathbb{R}) \times S(\mathbb{R}) \to \mathbb{R}$.

- Given a finite sequence $\mathbf{X} \in S(\mathbb{R}^N)$, dimensionality reduction takes place by determining the projection $\boldsymbol{\xi}^i$ of every $\boldsymbol{x}^i$ such that the costs

$$\mathrm{costs}(\mathbf{X\Xi}) := \sum_{\boldsymbol{x}^i \in \mathbf{X}} \mathrm{error}(\mathrm{char}_{\mathcal{X}}(\mathbf{X}, \boldsymbol{x}^i), \mathrm{char}_{\mathcal{E}}(\mathbf{X\Xi}, (\boldsymbol{x}^i, \boldsymbol{\xi}^i)) \tag{6.12}$$

  are minimized.

- Possibly, additional constraints are imposed on $\boldsymbol{\xi}^i$ to guarantee uniqueness or invariance of the result. This can be formalized by a constraint function

$$\mathrm{constraint} : S(\mathbb{R}^M \times \mathbb{R}^N) \to \mathbb{R} \tag{6.13}$$

  which is optimized simultaneously to the overall costs (6.12) and which can implement hard constraints by means of an indicator function or soft constraints by means of a real-valued function.

The methods differ in the definition of the data characteristics and in the way the error of the characteristics is defined. Furthermore, they differ in the (implicit or explicit) computation of the characteristics and the employed (analytical or numerical) optimization method. The objective (6.12) and the constraints (6.13) might be contradictory, and the way in which these two objectives are combined can be chosen differently.

Table 6.1 summarizes the properties of the different optimization methods with respect to this general principle. We explain the formalization and the exact choice of the relevant functions in more detail in the following:

MDS: the characteristics are the pairwise Euclidean distances in the original and embedding space respectively:

$$\mathrm{char}_{\mathcal{X}}(\mathbf{X}, \boldsymbol{x}) = (d_{\mathcal{X}}(\boldsymbol{x}^1, \boldsymbol{x}), \ldots, d_{\mathcal{X}}(\boldsymbol{x}^n, \boldsymbol{x}))$$

and

$$\mathrm{char}_{\mathcal{E}}(\mathbf{X}\Xi, (\boldsymbol{x}, \boldsymbol{\xi})) = (d_{\mathcal{E}}(\boldsymbol{\xi}^1, \boldsymbol{\xi}), \ldots, d_{\mathcal{E}}(\boldsymbol{\xi}^n, \boldsymbol{\xi}))$$

In particular, the characteristic $\mathrm{char}_{\mathcal{E}}$ depends on the projections of the data only and not the original coefficients in this case. The cost function is the least squared error, i.e.

$$\mathrm{error}((a_1, \ldots, a_n), (b_1, \ldots, b_n)) = \sum_{i=1}^{n} (a_i - b_i)^2 / a_i$$

for $a_i, b_i \in \mathbb{R}$, where the weighting corresponds to the Sammon mapping. Note that only sequences of the same length are compared via this function. No constraints are imposed, i.e. the constraint function (6.13) is trivial.

Isomap: Isomap differs from MDS only in the characteristic $\mathrm{char}_{\mathcal{X}}$ which is given by the geodesic distances $(d_{\mathrm{geodesic}}(\boldsymbol{x}^1, \boldsymbol{x}), \ldots, d_{\mathrm{geodesic}}(\boldsymbol{x}^n, \boldsymbol{x}))$. Geodesic distances are usually approximated in the data set by means of the following algorithm: A neighborhood graph is constructed from $\mathbf{X} = (\boldsymbol{x}^1, \ldots, \boldsymbol{x}^n)$ and $\boldsymbol{x}$ by means of an ϵ-neighborhood or a k-NN graph with vertices enumerated by $\boldsymbol{x}^i$ and $\boldsymbol{x}$. Then, all shortest paths from $\boldsymbol{x}$ to $\boldsymbol{x}^i$ are computed within this graph. These distances constitute an approximation of the geodesic distances of the underlying data manifold.

LLE: In LLE the characteristics are the local reconstruction weights of points estimated by their neighborhood, i.e.

$$\mathrm{char}_{\mathcal{X}}(\mathbf{X}, \boldsymbol{x}) = \mathrm{argmin}_{(w_1, \ldots, w_n)} \left\{ \left(\boldsymbol{x} - \sum_i 1_{\boldsymbol{x} \to \boldsymbol{x}^i} \, w_i \boldsymbol{x}^i \right)^2 \;\middle|\; \sum_i w_i = 1 \right\}$$

Table 6.1: Many dimensionality reduction methods can be put into a general framework: characteristics of the data are extracted. The low-dimensional coefficients lead to corresponding characteristics in the embedding space. These coefficients are determined such that an error measure of the characteristics is minimized, fulfilling probably additional constraints.

method	characteristics of data	characteristics of projections	error measure	constraint
MDS	Euclidean distance $d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{x}^j)$	Euclidean distance $d_{\mathcal{E}}(\boldsymbol{\xi}^i, \boldsymbol{\xi}^j)$	minimize (weighted) least squared error	no constraint
Isomap	Geodesic distance $d_{\text{geodesic}}(\boldsymbol{x}^i, \boldsymbol{x}^j)$	Euclidean distance $d_{\mathcal{E}}(\boldsymbol{\xi}^i, \boldsymbol{\xi}^j)$	minimize (weighted) least squared error	no constraint
LLE	weights w_{ij} such that $\sum(\boldsymbol{x}^i - \sum_{i \to j} w_{ij}\boldsymbol{x}^j)^2$ minimum with $\sum_j w_{ij} = 1$	weights $\tilde{w}_{ij}$ such that $\sum(\boldsymbol{\xi}^i - \sum_{i \to j} \tilde{w}_{ij}\boldsymbol{\xi}^j)^2$ is minimum	identity $w_{ij} = \tilde{w}_{ij}$	$\sum \boldsymbol{\xi}^i = 0$, $\sum_i \boldsymbol{\xi}^i (\boldsymbol{\xi}^i)^\top = n \cdot \mathbf{I}$
Laplacian Eigenmap	weights $w_{ij} = \exp(-d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{x}^j)^2/\sigma)$ for neighbors $i \to j$	distance $d_{\mathcal{E}}(\boldsymbol{\xi}^i, \boldsymbol{\xi}^j)^2$ for neighbors $i \to j$	minimize dot product	$\boldsymbol{\Xi}^\top \mathbf{D} \boldsymbol{\Xi} = \mathbf{I}$, $\boldsymbol{\Xi}^\top \mathbf{D} \mathbf{I} = \mathbf{0}$
MVU	Euclidean distance $d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{x}^j)$ for neighbors $i \to j$	Euclidean distance $d_{\mathcal{E}}(\boldsymbol{\xi}^i, \boldsymbol{\xi}^j)$ for neighbors $i \to j$	enforce identity using slack variables	maximize $\sum_{ij} d_{\mathcal{E}}(\boldsymbol{\xi}^i, \boldsymbol{\xi}^j)^2$ with $\sum_i \boldsymbol{\xi}^i = \mathbf{0}$.
SNE	probabilities $p_{j\|i} = \frac{\exp\left(-d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{x}^j)^2/2\sigma_i\right)}{\sum_{k \neq i} \exp\left(-d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{x}^k)^2/2\sigma_i\right)}$	probabilities $q_{j\|i} = \frac{\exp\left(-d_{\mathcal{E}}(\boldsymbol{\xi}^i, \boldsymbol{\xi}^j\right)^2}{\sum_{k \neq i} \exp\left(-d_{\mathcal{E}}(\boldsymbol{\xi}^i, \boldsymbol{\xi}^k)^2\right)}$	minimize Kullback-Leibler divergences	no constraint
t-SNE	probabilities $p_{ij} = \frac{p_{j\|i} + p_{i\|j}}{2n}$	probabilities $q_{ij} = \frac{(1 + d_{\mathcal{E}}(\boldsymbol{\xi}^i, \boldsymbol{\xi}^j)/\varsigma)^{-\frac{\varsigma+1}{2}}}{\sum_{k \neq l} (1 + d_{\mathcal{E}}(\boldsymbol{\xi}^k, \boldsymbol{\xi}^l)/\varsigma)^{-\frac{\varsigma+1}{2}}}$	minimize Kullback-Leibler divergence	no constraint
NeRV	probabilities $p_{j\|i}$ as for SNE	probabilities $q_{j\|i}$ as for SNE	minimize sum of Kullback-Leibler divergences with weight $c \in [0, 1]$	no constraint
t-NeRV	probabilities p_{ij} as for t-SNE	probabilities q_{ij} as for t-SNE	minimize sum of Kullback-Leibler divergences with weight $c \in [0, 1]$	no constraint

where $1_{\boldsymbol{x}\to\boldsymbol{x}^i}$ denotes the characteristic function of the neighbors of $\boldsymbol{x}$ in $\mathbf{X}$, excluding $\boldsymbol{x}$ itself.

$$\mathrm{char}_{\mathcal{E}}(\mathbf{X}\Xi,(\boldsymbol{x},\boldsymbol{\xi})) = \mathrm{argmin}_{(\tilde{w}_1,\ldots,\tilde{w}_n)} \left\{ \left(\boldsymbol{\xi} - \sum_i 1_{\boldsymbol{x}\to\boldsymbol{x}^i}\ \tilde{w}_i \boldsymbol{\xi}^i \right)^2 \right\}$$

This characteristic uses both, the projections $\boldsymbol{\xi}^i$, and the data in original space $\boldsymbol{x}^i$ to define the neighborhood graph. Since the characteristic $\mathrm{char}_{\mathcal{E}}$ already includes an approximation, the error can be picked in a trivial way:

$$\mathrm{error}((a_1,\ldots,a_n),(b_1,\ldots,b_n)) = \begin{cases} 0 & \text{if } \forall i\ a_i = b_i \\ 1 & \text{otherwise} \end{cases}$$

Because of this definition, minimization of (6.12) is equivalent to a minimization of $\sum_i (\boldsymbol{\xi}^i - \sum_j 1_{\boldsymbol{x}^i\to\boldsymbol{x}^j}\ w_{ij}\boldsymbol{\xi}^j)^2$ where the reconstruction weights w_{ij} and the neighborhood structure $1_{\boldsymbol{x}^i\to\boldsymbol{x}^j}$ are taken from the original data space. Since this formulation is not well posed, 0 being an obvious global optimum, regularization is used. The constraints enforce that the projection coefficients are centered at the origin and their correlation matrix is given by the unit matrix. Since these constraints can be fulfilled exactly, the characteristic function

$$\mathrm{constraint}(\mathbf{X}\Xi) = \begin{cases} 0 & \text{if } \sum \boldsymbol{\xi}^i = \mathbf{0} \text{ and } \sum_i \boldsymbol{\xi}^i (\boldsymbol{\xi}^i)^\top = n \cdot \mathbf{I} \\ 1 & \text{otherwise} \end{cases}$$

can be used.

Laplacian Eigenmap: The characteristics of the original data space is based on the local neighborhood structure and an appropriate weighting of distances given in this neighborhood, e.g. weighting according to the heat kernel:

$$\mathrm{char}_{\mathcal{X}}(\mathbf{X},\boldsymbol{x}) = (1_{\boldsymbol{x}\to\boldsymbol{x}^1}\cdot\exp(-(\boldsymbol{x}-\boldsymbol{x}^1)^2/\sigma),\ldots,1_{\boldsymbol{x}\to\boldsymbol{x}^n}\cdot\exp(-(\boldsymbol{x}-\boldsymbol{x}^n)^2/\sigma))\ .$$

Characteristics of the projections are similar, but based on the standard Euclidean distance

$$\mathrm{char}_{\mathcal{E}}(\mathbf{X}\Xi,(\boldsymbol{x},\boldsymbol{\xi})) = (1_{\boldsymbol{x}\to\boldsymbol{x}^1}\cdot(\boldsymbol{\xi}-\boldsymbol{\xi}^1)^2,\ldots,1_{\boldsymbol{x}\to\boldsymbol{x}^n}\cdot(\boldsymbol{\xi}-\boldsymbol{\xi}^n)^2)\ .$$

The cost function is given by the dot product:

$$\mathrm{error}((a_1,\ldots,a_n),(b_1,\ldots,b_n)) = \sum_i a_i b_i$$

which is minimized. Since this formulation allows the trivial solution 0, constraints are imposed. Set $d_{ii} = \sum_j 1_{\boldsymbol{x}^i \to \boldsymbol{x}^j} \exp(-(\boldsymbol{x}^i - \boldsymbol{x}^j)^2/\sigma)$, then an arbitrary scaling factor and translation of the solution is removed by imposing the constraint function

$$\text{constraint}(\mathbf{X}\boldsymbol{\Xi}) = \begin{cases} 0 & \text{if } \sum_i d_{ii}\boldsymbol{\xi}^i(\boldsymbol{\xi}^i))^\top = \mathbf{I} \text{ and } \sum_i d_{ii}\boldsymbol{\xi}^i = \mathbf{0} \\ 1 & \text{otherwise} \end{cases}$$

MVU: Similarly,

$$\text{char}_{\mathcal{X}}(\mathbf{X}, \boldsymbol{x}) = (1_{\boldsymbol{x}\to\boldsymbol{x}^1} \cdot (\boldsymbol{x} - \boldsymbol{x}^1)^2, \ldots, 1_{\boldsymbol{x}\to\boldsymbol{x}^n} \cdot (\boldsymbol{x} - \boldsymbol{x}^n)^2)$$

and

$$\text{char}_{\mathcal{E}}(\mathbf{X}\boldsymbol{\Xi}, (\boldsymbol{x}, \boldsymbol{\xi})) = (1_{\boldsymbol{x}\to\boldsymbol{x}^1} \cdot (\boldsymbol{\xi} - \boldsymbol{\xi}^1)^2, \ldots, 1_{\boldsymbol{x}\to\boldsymbol{x}^n} \cdot (\boldsymbol{\xi} - \boldsymbol{\xi}^n)^2)$$

with error

$$\text{error}((a_1, \ldots, a_n), (b_1, \ldots, b_n)) = \begin{cases} 0 & \text{if } \forall i\ a_i = b_i \\ 1 & \text{otherwise} \end{cases}$$

and constraint

$$\text{constraint}(\mathbf{X}\boldsymbol{\Xi}) = -\sum_{ij}(\boldsymbol{\xi}^i - \boldsymbol{\xi}^j)^2 + \begin{cases} 0 & \text{if } \sum_i \boldsymbol{\xi}^i = \mathbf{0} \\ \\ c & \text{otherwise} \end{cases}$$

with a constant c. The cost term defines a characteristic function which might not possess a feasible solution because it is in general not possible to exactly preserve all local distances. Therefore, the cost function should be "smoothed". In MVU, the characteristic functions are taken as constraints of an optimization problem and slack variables are introduced.

SNE: Similarly,

$$\text{char}_{\mathcal{X}}(\mathbf{X}, \boldsymbol{x}) = \left[\frac{\exp\left(\frac{-d_{\mathcal{X}}(\boldsymbol{x}, \boldsymbol{x}^i)^2}{2\sigma_{\boldsymbol{x}}}\right)}{\sum_{\boldsymbol{x}^k \neq \boldsymbol{x}} \exp\left(\frac{-d_{\mathcal{X}}(\boldsymbol{x}, \boldsymbol{x}^k)^2}{2\sigma_{\boldsymbol{x}}}\right)} \right]_{i=1,\ldots,n}$$

where entries corresponding to $\boldsymbol{x}^i = \boldsymbol{x}$ are set to 0, and

$$\text{char}_{\mathcal{E}}(\mathbf{X}\boldsymbol{\Xi}, (\boldsymbol{x}, \boldsymbol{\xi})) = \left[\frac{\exp\left(-d_{\mathcal{E}}(\boldsymbol{\xi}, \boldsymbol{\xi}^i\right)^2}{\sum_{\boldsymbol{x}^k \neq \boldsymbol{x}} \exp\left(-d_{\mathcal{E}}(\boldsymbol{\xi}, \boldsymbol{\xi}^k)^2\right)} \right]_{i=1,\ldots,n}$$

again setting entries for $\boldsymbol{\xi}^i = \boldsymbol{\xi}$ to 0. The bandwidth parameter $\sigma_{\boldsymbol{x}}$ is determined such that the effective number of neighbors of $\boldsymbol{x}$ in $\mathbf{X}$ as measured via an information theoretic framework is equal to a predefined value, the perplexity, which constitutes a meta-parameter of the model. The error is given by the Kullback Leibler divergence

$$\text{error}((a_1, \ldots, a_n), (b_1, \ldots, b_n)) = \sum_i a_i \log \frac{a_i}{b_i}$$

No constraints are imposed.

t-SNE: Similar to SNE, we have

$$\text{char}_{\mathcal{X}}(\mathbf{X}, \boldsymbol{x}) = 1/(2(|\mathbf{X} \cup \{\boldsymbol{x}\}|)) \cdot \left[\frac{\exp\left(\frac{-d_{\mathcal{X}}(\boldsymbol{x}, \boldsymbol{x}^i)^2}{2\sigma_{\boldsymbol{x}}}\right)}{\sum_{\boldsymbol{x}^k \in \mathbf{X}, \boldsymbol{x}^k \neq \boldsymbol{x}} \exp\left(\frac{-d_{\mathcal{X}}(\boldsymbol{x}, \boldsymbol{x}^k)^2}{2\sigma_{\boldsymbol{x}}}\right)} \right]_{i=1,\ldots,n}$$
$$+ 1/(2(|\mathbf{X} \cup \{\boldsymbol{x}\}|)) \cdot \left[\frac{\exp\left(\frac{-d_{\mathcal{X}}(\boldsymbol{x}, \boldsymbol{x}^i)^2}{2\sigma_{\boldsymbol{x}^i}}\right)}{\sum_{\boldsymbol{x}^k \in \mathbf{X} \cup \{\boldsymbol{x}\}, \boldsymbol{x}^k \neq \boldsymbol{x}^i} \exp\left(\frac{-d_{\mathcal{X}}(\boldsymbol{x}^k, \boldsymbol{x}^i)^2}{2\sigma_{\boldsymbol{x}^i}}\right)} \right]_{i=1,\ldots,n}$$

where $\mathbf{X} \cup \{\boldsymbol{x}\}$ refers to the set of elements without duplicates, and

$$\text{char}_{\mathcal{E}}(\mathbf{X}\boldsymbol{\Xi}, (\boldsymbol{x}, \boldsymbol{\xi})) = \left[\frac{(1 + (\boldsymbol{\xi} - \boldsymbol{\xi}^i)^2)^{-1}}{\sum_{\boldsymbol{x}^k \neq \boldsymbol{x}^l \in \mathbf{X} \cup \{\boldsymbol{x}\}} (1 + (\boldsymbol{\xi}^k - \boldsymbol{\xi}^l)^2)^{-1}} \right]_{i=1,\ldots,n}$$

setting entries corresponding to $\boldsymbol{x} = \boldsymbol{x}^i$ to 0. Again, the Kullback Leibler divergence is used and no constraints are imposed.

NeRV: NeRV deviates from SNE only in the choice of the cost function which is

$$\text{error}((a_1, \ldots, a_n), (b_1, \ldots, b_n)) = c \sum_i a_i \log \frac{a_i}{b_i} + (1 - c) \sum_i b_i \log \frac{b_i}{a_i}$$

with appropriate weighting c.

t-NeRV: Similarly, t-NeRV uses the same cost function as NeRV in the t-SNE setting.

These formalizations are summarized in Tab. 6.1. Note that some of the techniques allow for an explicit algebraic solution or lead to a unique optimum such as LLE, MVU, and Laplacian eigenmaps, while others require numeric optimization

such as SNE and its variants. For the latter cases, unique solutions usually do not exist and multiple local optima may be found depending on the initialization of the parameters. Visualizations obtained this way can differ significantly from one run to the next depending on the initialization strategy. However, as argued by (van der Maaten and Hinton 2008), this fact is not necessarily a drawback of the technique. Usually, high-dimensional data sets cannot be embedded into low-dimensions without loss of information. Often, there exists more than one reasonable embedding of data which is inherently ambiguous. Different local optima of the projection techniques can correspond to different low-dimensional views of the data with the same quality (as measured e.g. using evaluation measures as proposed by (Lee and Verleysen 2009, Venna et al. 2010)). This argument is in line with our experimental observation, that dimension reduction based on t-SNE leads to qualitatively different behavior in different runs. However, the quality of the different results usually does not differ much from each other when using the quality measure proposed by (Lee and Verleysen 2009), for instance.

6.2.2 Out-of-sample extensions

One benefit of our general formulation is that the optimization steps are separated from the principled mathematical objective of the actual technique at hand. As an immediate consequence, a principled framework for out-of-sample extension can be formalized simultaneously for all techniques. Here, out-of-sample extension refers to the question of how to extend the projection to a novel point $\boldsymbol{x} \in \mathbb{R}^N$ if a set of points $\mathbf{X}$ is already mapped to projections $\boldsymbol{\Xi}$. Assume that a dimension reduction for a given data set is given, characterized by the sequence of points and their projections $\mathbf{X}\boldsymbol{\Xi}$. Assume that a novel data point $\boldsymbol{x}$ is considered. Then, a reasonable projection $\boldsymbol{\xi}$ of this point can be obtained by means of the mapping: $\boldsymbol{x} \mapsto \boldsymbol{\xi}$ such that the costs

$$\mathrm{error}(\mathrm{char}_{\mathcal{X}}(\mathbf{X}, \boldsymbol{x}), \mathrm{char}_{\mathcal{E}}(\mathbf{X}\boldsymbol{\Xi}, (\boldsymbol{x}, \boldsymbol{\xi}))$$

are minimized. This term corresponds to the contribution of $\boldsymbol{x}$ and its projection $\boldsymbol{\xi}$ to the overall costs (6.12) assuming that the projections $\boldsymbol{\Xi}$ of $\mathbf{X}$ are fixed. Simultaneously, the constraints

$$\mathrm{constraint}(\mathbf{X}\boldsymbol{\Xi} \bullet (\boldsymbol{x}, \boldsymbol{\xi}))$$

need to be optimized where $\mathbf{X}\boldsymbol{\Xi} \bullet (\boldsymbol{x}, \boldsymbol{\xi})$ denotes the concatenation of the known coordinates and the novel projection $(\boldsymbol{x}, \boldsymbol{\xi})$, where again, the coefficients $\boldsymbol{\Xi}$ are kept fixed and only the novel projection coordinates $\boldsymbol{\xi}$ are treated as free parameters. For simple constraints such as given for MDS, Isomap, and SNE and its variants, this immediately yields a mathematical formalization of out-of-sample extensions. Numerical optimization such as gradient techniques can be used to obtain solutions.

For LLE and Laplacian Eigenmaps the constraints are given by an indicator function, the same holds for the constraint $\sum \boldsymbol{\xi}^i = 0$ for MVU. These constraints can no longer exactly be fulfilled and should be weakened to soft constraints. This has the consequence that, in general, explicit algebraic solutions of the optimization problem are no longer available.

Typically, the complexity of this approach depends on the number n of the given data points. Hence, this procedure can be quite time consuming depending on the given data set. Moreover, this mapping leads to an implicit functional prescription in terms of an optimum of a complicated function, which may display local optima.

In the following, we will substitute the implicit form by an explicit functional prescription the form of which is fixed a priori. We derive techniques to determine function parameters by means of the given optimization objectives. The fact that non-parametric dimensionality reduction is formalized via a general framework allows us to simultaneously extend all these methods to explicit mapping functions in a principled way.

6.3 Dimension reduction mappings

Due to their dependency on pairwise dissimilarities, the computational effort of most dimensionality reduction techniques scales quadratically with respect to the number of data points. This makes them infeasible for large data sets. Even linear techniques, such as presented in (Bunte, Hammer, Villmann, Biehl and Wismüller 2011), can reach their limits for very large data sets so that sub-linear or even constant time techniques are required. Furthermore, it might be inadequate to display all data points given a large data set due to the limited resolution on screens or prints. Therefore, in the literature, often a random subsample of the full data set is picked as representative of the data, see e.g. the overviews (van der Maaten et al. 2009, Venna et al. 2010). If additional points are added on demand, out-of-sample extension as specified above is necessary.

One crucial property of this procedure consists in the requirement that the mapping which is determined from a small subsample is representative for a mapping of the full data set. Hence, the generalization ability of dimensionality reduction to novel data points must be guaranteed. To our knowledge, the generalization ability of non-parametric dimension reduction has hardly been verified experimentally in the literature (one exception being presented e.g. by (Suykens 2008)), nor do exact mathematical treatments of the generalization ability exist.

Here, we take a different point of view and address the problem of dimensionality reduction by inferring an explicit mapping function. This has several benefits:

a mapping function allows immediate extension to novel data points by simply applying the mapping. Hence, large data sets can be dealt with since the mapping function can be inferred from a small subset only in constant time (assuming constant size of the subset). Mapping all data points requires linear time only. The generalization ability of the mapping function can be addressed explicitly in experiments. We will observe an excellent generalization ability in several examples. Furthermore, the generalization ability can be treated in an exact mathematical way by referring to the mapping function. We will argue that for typical mapping functions guarantees exist in the framework of statistical learning theory. An additional benefit consists in the fact that the complexity of the mapping function and its functional form can be chosen priorly, such that auxiliary information, e.g. in terms of class labels, can be integrated into the system.

6.3.1 Previous work

A few dimensionality reduction techniques provide an explicit mapping of the data: Linear methods such as PCA or neighborhood preserving projection optimize the information loss of the projection (Bishop 2006, He et al. 2005). Extensions to nonlinear functions are given by autoencoder networks, which provide a function given by a multilayer feedforward network in such a way that the reconstruction error is minimized when projecting back with another feedforward network (van der Maaten et al. 2009). Typically, training is done by standard back propagation, directly minimizing the reconstruction error. Manifold charting connects local linear embeddings obtained by local PCA, for example, by minimizing the error on the overlaps (Brand 2002, Teh and Roweis 2003). This can be formulated in terms of a generalized eigenvalue problem. Topographic maps such as the self-organizing map or generative topographic mapping characterize data in terms of prototypes which are visualized in low-dimensions (Bishop and Williams 1998, Kohonen et al. 2001). Due to the clustering, new data points can directly be visualized by mapping them to the closest prototype or its visualization, respectively.

Some non-parametric dimension reduction methods, as introduced above, have been extended to global dimension reduction mappings. For example, LLC (Teh and Roweis 2003) extends LLE by assuming that local linear projections are available, such as local PCAs, and combining these using affine transformations. The resulting points are inserted in the LLE cost function and additional parameters are optimized accordingly. Kernel maps, based on the ideas of kernel eigenmap methods, provide direct out-of-sample extensions with excellent generalization ability (Suykens 2008). Parametric t-SNE (van der Maaten 2009) extends t-SNE towards an embedding given by a multilayer neural network. The network parameters are

determined using back propagation, where, instead of the mean squared error, the t-SNE cost function is taken as objective. These techniques, however, are often specifically tailored to the functional form of the mapping or the specific properties of the technique. In contrast, we propose a general principle to extend non-parametric dimension reduction to explicit mappings.

6.3.2 A general principle

As explained above, a dimension reduction technique determines an implicit function of the full data space to the projection space $f : \mathbb{R}^N \to \mathbb{R}^M$. A data point $\boldsymbol{x}$ is projected to low-dimensional counterparts which minimizes the respective cost function and constraints. Depending on the method, f might have a complex form and its computation might be time consuming. This computational complexity can be avoided by defining an explicit dimension reduction mapping function:

$$f_W : \mathbb{R}^N \to \mathbb{R}^M, \boldsymbol{x} \to \widehat{\boldsymbol{\xi}} = f_W(\boldsymbol{x}) \tag{6.14}$$

of fixed form parameterized by W. The general formalization of dimension reduction as cost optimization allows us to extend non-parametric embedding to an explicit mapping function f_W as follows: We fix a parameterized function $f_W : \mathbb{R}^N \to \mathbb{R}^M$. Instead of the projection coordinates $\boldsymbol{\xi}$, we consider the images of the mapping $\widehat{\xi} = f_W(\boldsymbol{x})$ and optimize the parameters W such that the costs

$$\text{costs}(\mathbf{X}\widehat{\boldsymbol{\Xi}}) = \sum_{\boldsymbol{x}^i \in \mathbf{X}} \text{error}(\text{char}_{\mathcal{X}}(\mathbf{X}, \boldsymbol{x}^i), \text{char}_{\mathcal{E}}(\mathbf{X}\widehat{\boldsymbol{\Xi}}, (\boldsymbol{x}^i, \widehat{\boldsymbol{\xi}}^i))) \tag{6.15}$$

become minimal, under the constraints

$$\text{constraints}(\mathbf{X}\widehat{\boldsymbol{\Xi}}) \tag{6.16}$$

where $\mathbf{X}\widehat{\boldsymbol{\Xi}}$ refers to the sequence $((\boldsymbol{x}^1, \widehat{\boldsymbol{\xi}}^1 = f_W(\boldsymbol{x}^1), \ldots, (\boldsymbol{x}^n, \widehat{\boldsymbol{\xi}}^n = f_W(\boldsymbol{x}^n))$.

This principle leads to a well defined mathematical objective for the mapping parameters W for every dimension reduction method as summarized in Tab. 6.1. For out-of-sample extensions, however, hard constraints such as imposed for LLE, MVU, and Laplacian eigenmaps can no longer exactly be fulfilled and should be transferred to soft constraints. This has the consequence that the optimization problem differs from the on in the original method: A closed form solution as given for, e.g. spectral methods might no longer be available for a general functional form f_W and soft constraints. The functional form f_W need to be specified a priori. It can be chosen as a global linear function, a combination of locally linear projections, a feedforward neural network, or any parameterized, possibly non-linear, function. If

gradient techniques are used for the optimization of the parameters W, f_W has to be differentiable with respect to W. The functional form of f_W defines the flexibility of the resulting dimensionality reduction mapping. Naturally, restricted choices such as linear forms lead to less flexibility than universal approximators such as feedforward networks or general kernel maps.

Note that this provides a general framework which extends dimensionality reduction techniques in order to obtain explicit mapping functions. The ingredients are formally defined for all methods specified in Table 6.1. This gives a mathematical objective for all functional forms of f_W and all these methods, provided hard constraints of LLE and similar are softened in such a way that feasible solutions result. The objectives can directly be optimized using universal optimization techniques such as gradient methods or local search techniques. Explicit algebraic solutions as given for the original spectral techniques are no longer available, however. Furthermore, the numeric optimization task can be difficult in practice.

Since every possible dimension reduction techniques and every choice of the form f_W leads to a different method, an extensive evaluation of all possible choices is beyond the scope of this thesis. In the next section we consider example algorithms for two specific mapping functions: a global linear one and a non-linear mapping based on local linear projections in the t-SNE formalism. For the latter setting, we first demonstrate the feasibility of the results in the unsupervised setting for local linear maps in comparison to feedforward networks used for dimension reduction. Then, we demonstrate the possibility to integrate supervised label information into the technique by means of a bias of the functional form of f_W.

6.4 Linear t-SNE mapping

In this section we derive the formulation based on a linear hypothesis for the mapping, optimized according to the t-SNE cost function. In this case the mapping

$$f_W : \boldsymbol{x}^i \rightarrow \widehat{\boldsymbol{\xi}}^i = A \cdot \boldsymbol{x}^i \tag{6.17}$$

is expressed in terms of a rectangular matrix A which defines a linear transformation from $\mathbb{R}^N \rightarrow \mathbb{R}^M$. This matrix can be optimized by following a stochastic gradient descent procedure using the gradient of the t-SNE cost function (Eq. (6.8)):

$$\begin{aligned}\frac{\partial E_{\mathrm{t-SNE}}}{\partial A} =& \sum_i \sum_j \frac{\partial E_{\mathrm{t-SNE}}}{\partial q_{ij}} \cdot \frac{\partial q_{ij}}{\partial d_{\mathcal{E}}(\widehat{\boldsymbol{\xi}}^i, \widehat{\boldsymbol{\xi}}^j)^2} \cdot \frac{\partial d_{\mathcal{E}}(\widehat{\boldsymbol{\xi}}^i, \widehat{\boldsymbol{\xi}}^j)^2}{\partial A} \\ =& \frac{\varsigma + 1}{2\varsigma} \sum_i \sum_j (p_{ij} - q_{ji}) \cdot \left(1 + \frac{d_{\mathcal{E}}(\widehat{\boldsymbol{\xi}}^i, \widehat{\boldsymbol{\xi}}^j)}{\varsigma}\right)^{-1} \cdot \frac{\partial d_{\mathcal{E}}(\widehat{\boldsymbol{\xi}}^i, \widehat{\boldsymbol{\xi}}^j)^2}{\partial A}\end{aligned}$$

Using Euclidean distance $d_{\mathcal{E}}(\widehat{\boldsymbol{\xi}}^i, \widehat{\boldsymbol{\xi}}^j) = ||A\boldsymbol{x}^i - A\boldsymbol{x}^j||$ it follows:

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

and hence (see Appendix 6.A.1 for details)

$$\frac{\partial E_{\mathrm{t-SNE}}}{\partial A} = \frac{\varsigma + 1}{\varsigma} \sum_i \sum_j \frac{(p_{ij} - q_{ji})}{1 + \frac{1}{\varsigma}||A\boldsymbol{x}^i - A\boldsymbol{x}^j||^2} \cdot (A\boldsymbol{x}^i - A\boldsymbol{x}^j)(\boldsymbol{x}^i - \boldsymbol{x}^j) \ . \tag{6.18}$$

An example result of this algorithm on a three dimensional benchmark data set is compared to simple PCA. The data contains three Gaussians arranged on top of each other (see upper left panel of Figure 6.1). Because of the variance in the z-direction PCA projects the modes onto each other loosing the cluster information (see lower left panel in Figure 6.1). The linear mapping obtained by the optimization of the t-SNE cost function (referred to as DiReduct mapping) on the other hand shows a much clearer separation of the original clusters (see upper right panel of Figure 6.1). This is due to the preservation of local structures formulated in the t-SNE objective rather than the preservation of global variances as used in PCA.

A quantitative evaluation of the two mappings is also included in the lower right panel of Figure 6.1, based on the quality measure proposed by (Lee and Verleysen 2008, Lee and Verleysen 2009). Basically, it relies on k-intrusions and k-extrusions, which means it compares k-ary neighborhoods given in the original high-dimensional space with those occurring in the low-dimensional space. Intrusion refers to samples intruding a neighborhood in the embedding space, while extrusion counts the number of samples which are missing in the projected k-ary neighborhoods. The overall quality measure Q measures the percentage of data which is neither k-intrusive nor k-extrusive. In the optimal case all neighborhoods are exactly preserved which results in a value of $Q = 1$. B measures the percentage of k-intrusions minus the percentage of k-extrusions in the projection and therefore shows the tendency of the mapping method: techniques with negative values for B are characterized by extrusive behavior, while those with positive values tend to be more intrusive. The procedure is summarized in Algorithm 6.6:

Obviously, DiReduct shows a superior quality, in particular for small neighborhood ranges, since it preserves local structures of the data to a larger extent. Further, unlike PCA which displays a trend towards highly intrusive behavior, it is rather neutral in the mapping character, being mildly extrusive for medium values of k.

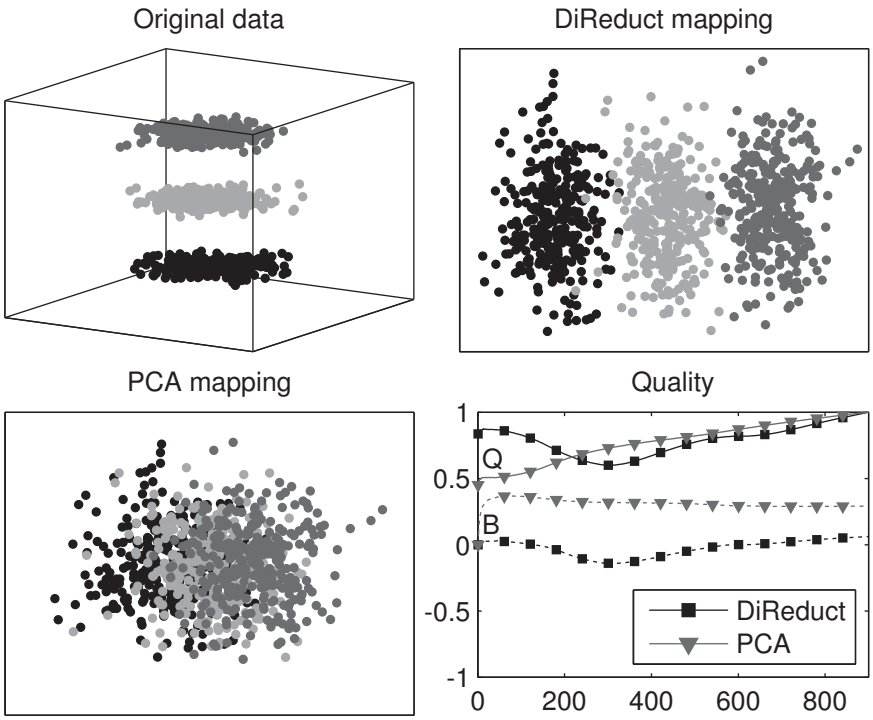


Figure 6.1: Simulation results on a three class benchmark data set using PCA and a global linear map optimizing the t-SNE cost function, respectively. The latter leads to a better separation due to its local nature, which can be formally evaluated using the measure of intrusion and extrusion on the resulting mapping.

Algorithm 6.6 : Intrusion / Extrusion measure for dimension reduction

1: compute the co-ranking matrix $\mathbf{Q} = [\,|\,\{(i,j) : R_{ij} = k \text{ and } \mathcal{R}_{ij} = l\}\,|\,]_{1 \leqslant k,l \leqslant n-1}$ with R_{ij} and $\mathcal{R}_{ij}$ denoting the rank of sample $\boldsymbol{x}^i$ with respect to $\boldsymbol{x}^j$ and the rank of $\boldsymbol{\xi}^i$ with respect to $\boldsymbol{\xi}^j$ in the high- and low-dimensional space, respectively
2: use blocks of the co-ranking matrix to identify k-intrusions and k-extrusions
3: obtain the overall quality Q based on the weighted averages that take into account all k-intrusions and k-extrusions
4: compute B measuring the percentage of k-intrusions minus the percentage of k-extrusions indicating the overall behavior of the dimension reduction: negative values imply extrusive and positive values indicate intrusive behavior

6.5 Local Linear t-SNE mappings

In this section we consider non-linear mapping functions obtained by the principles outlined above. Again, we employ the t-SNE cost function. The functional form f_W is chosen in two different ways: First, we consider f_W given by a multilayer feedforward network as proposed by (van der Maaten 2009). The update equations for a feedforward network can be derived from the t-SNE cost function and are similar to standard back-propagation, see (van der Maaten 2009) for details.

Second, we consider a locally linear projection which is based on local mappings obtained by prototype-based techniques such as Neural Gas (NG) in combination with local PCA or mixtures of probabilistic PCA (Möller and Hoffmann 2004). The latter techniques provide a set of prototypes $\boldsymbol{w}^k \in \mathbb{R}^N$, dividing the data space into k receptive fields, and corresponding local projections $\Omega^k \in \mathbb{R}^{m \times N}$ with $m \leqslant N$. We assume that locally linear projections of the data points are derived from one of these techniques:

$$\boldsymbol{x}^i \to p^k(\boldsymbol{x}^i) = \Omega^k(\boldsymbol{x}^i - \boldsymbol{w}^k) \tag{6.19}$$

with local matrices Ω^k and prototypes $\boldsymbol{w}^k$. We assume furthermore the existence of responsibilities r_{ik} of the local mapping p^k for data point $\boldsymbol{x}^i$, where $\sum_k r_{ik} = 1$. In the following, we choose simple responsibilities based on the receptive fields:

$$r_{ik} = \begin{cases} 1 & \text{if } d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{w}^k) \leqslant d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{w}^j) \ \forall k \neq j \\ 0 & \text{otherwise} \end{cases} \tag{6.20}$$

More generally, a point $\boldsymbol{x}$ is associated with the responsibilities $r_k(\boldsymbol{x})$ in the same way. A global non-linear mapping function combines these linear projections:

$$f_W : \boldsymbol{x} \to \widehat{\boldsymbol{\xi}} = \sum_k r_k (A^k \cdot p^k(\boldsymbol{x}) + \boldsymbol{o}^k) \ , \tag{6.21}$$

using local linear projections $A^k \in \mathbb{R}^{M \times m}$ with $M \leqslant m$ and local offsets $\boldsymbol{o}^k \in \mathbb{R}^M$ to align the local pieces. The number of parameters W that have to be determined, depends on the number of local projections k and their dimension M. Usually, it is much smaller than the number of parameters when projecting all points $\boldsymbol{\xi}^i$ directly. Hence, it is sufficient to consider a small part of the given training data only, in order to obtained a valid dimension reduction. We determine the parameters by a stochastic gradient descent based on the derivative of the t-SNE cost function (see also Appendix 6.A.2):

$$\frac{\partial E_{\mathrm{t-SNE}}}{\partial \boldsymbol{o}^k} = \frac{\varsigma + 1}{\varsigma} \sum_{ij} \frac{(p_{ij} - q_{ji})}{1 + \frac{1}{\varsigma} d_{\mathcal{E}}(\widehat{\boldsymbol{\xi}}^i, \widehat{\boldsymbol{\xi}}^j)^2} \cdot (\widehat{\boldsymbol{\xi}}^i - \widehat{\boldsymbol{\xi}}^j)(r_{ik} - r_{jk}) \tag{6.22}$$

and

$$\frac{\partial E_{\mathrm{t-SNE}}}{\partial A^k} = \frac{\varsigma+1}{\varsigma} \sum_{ij} \frac{(p_{ij}-q_{ji})}{1+\frac{1}{\varsigma} d_{\mathcal{E}}(\widehat{\boldsymbol{\xi}}^i, \widehat{\boldsymbol{\xi}}^j)^2} \cdot (\widehat{\boldsymbol{\xi}}^i - \widehat{\boldsymbol{\xi}}^j)(r_{ik} p^k(\boldsymbol{x}^i) - r_{jk} p^k(\boldsymbol{x}^j)) \quad (6.23)$$

assuming Euclidean distance in the projection space, as before.

As an example, we show the results obtained on the UCI image segmentation data set. It consists of 7 classes and 2310 instances of 3×3 regions randomly drawn from 7 hand segmented outdoor images. Three of the 19 features were not taken into account, because they show no variance. We scaled the features by dividing with the maximal feature value in the data followed by PCA reducing the dimension to $m = 10$. For the locally linear projection, we run the NG algorithm (Martinetz and Schulten 1991, Cottrell et al. 2006) with 14 prototypes to get a division of the data space into receptive fields. PCA was applied to every receptive field to define local transformations Ω^k. Together with the respective prototypes w^k this offers the corresponding data projections $p^k(\boldsymbol{x}^i)$ (see Eq. (6.19)). The transformations $A^k \in \mathbb{R}^{2\times 10}$ were set as rectangular matrices to perform the dimension reduction from 10 to 2 dimensions. The offsets $\boldsymbol{o}^k$ are vectors in $\mathbb{R}^2$. The mapping parameters were initialized with small random values and a stochastic gradient descent was performed with $t_{\max} = 300$ epochs and learning rate

$$\tau_1(t) = \tau_1^{\mathrm{start}} \cdot \exp\left(-\frac{\log\left(\frac{\tau_1^{\mathrm{start}}}{\tau_1^{\mathrm{end}}}\right) t}{t_{\max}}\right) \quad (6.24)$$

$$\tau_2(t) = \tau_2^{\mathrm{start}} \cdot \exp\left(-\frac{\log\left(\frac{\tau_2^{\mathrm{start}}}{\tau_2^{\mathrm{end}}}\right) t}{t_{\max}}\right) \quad (6.25)$$

annealed from $\tau_1^{\mathrm{start}} = \tau_2^{\mathrm{start}} = 0.3$ to $\tau_1^{\mathrm{end}} = \tau_2^{\mathrm{end}} = 0.01$. The perplexity of t-SNE was set to 50. For the neural network embedding, we use parametric t-SNE with default parameters as provided in the implementation given by (van der Maaten 2009). An optimum network architecture was picked varying the number of neurons from 50 to 2000 per hidden layer. The architecture is given by a $[100\ 100\ 500\ 2]$-layer neural network. The perplexity was optimized on the data and picked as 25.

The results for a locally linear t-SNE mapping and parametric t-SNE are shown in Figure 6.2. In both cases, we used a subset of roughly ten percent for training, and we report the results of the mapping on training set and test set. Since the data set is labeled, an evaluation of the projection in terms of the nearest neighbor classification error is possible. The 5 nearest neighbor error for the whole preprocessed data after PCA to 10 dimensions is 0.054. After further dimension reduction this error increase

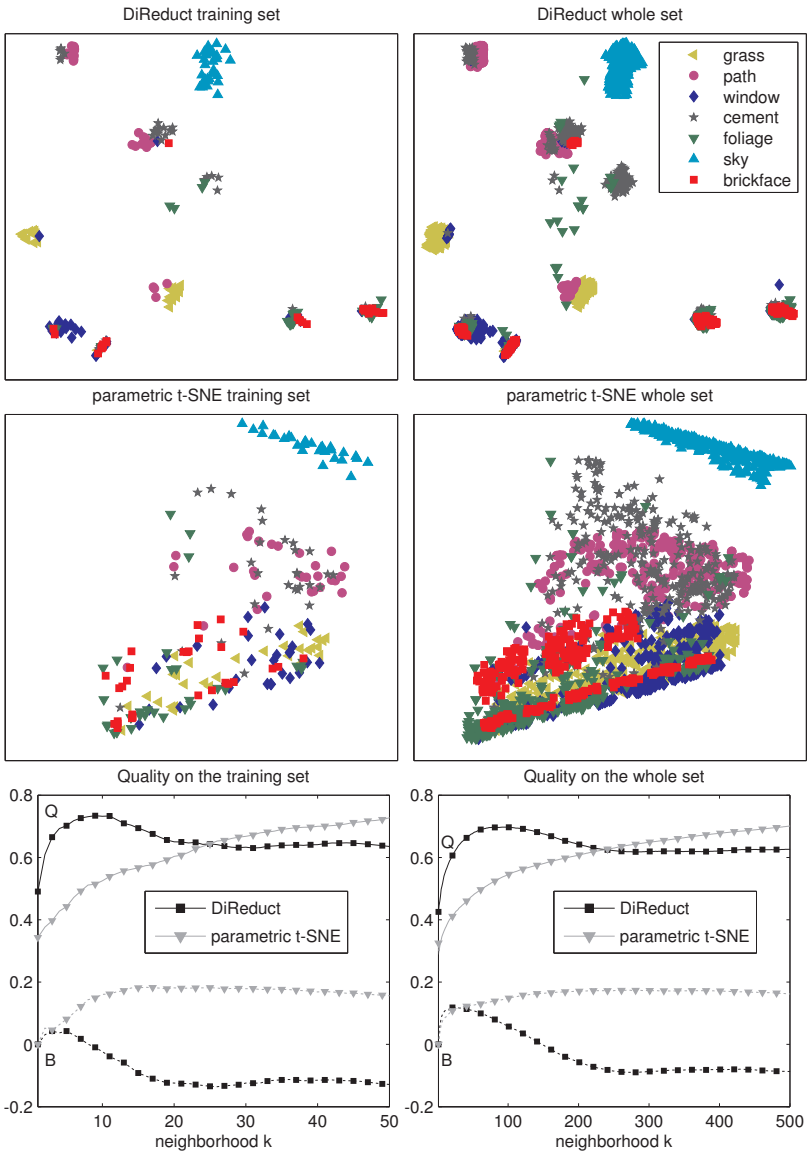


Figure 6.2: Projection of the UCI image segmentation data set using parametric t-SNE and DiReduct combining unsupervised clustering and the learning of a mapping. The result of the subsample used for training (left panels) as well as the full data set (right panels) are depicted. The intrusion/extrusion quality on the whole data set for both methods is shown in the bottom row.

due to the loss of information. For a locally linear mapping the 5-nearest neighbor error is 0.21 for the training set and 0.16 for the full data set, the corresponding projections are shown in the upper panel. The panels in the middle show the corresponding mappings achieved by parametric t-SNE (5 nearest neighbor error: 0.5 in training and 0.32 for the whole set, respectively). The bottom panel contains the evaluation of the mappings using the quality measure depicted in Algorithm 6.6 as proposed by (Lee and Verleysen 2008, Lee and Verleysen 2009). Interestingly, both functional forms show a good generalization ability in the sense that the error of the full data set resembles the error on the test set. However, the results of locally linear mappings are superior to a feedforward mapping in both cases.

6.6 Supervised dimensionality reduction mapping

Mapping high-dimensional data to low-dimensions is connected to an information loss and, depending on the dimension reduction technique, different data visualizations are derived. Since many methods such as t-SNE do not yield a unique solution, it can even happen that a data set is visualized in different ways with a single visualization technique in different runs. It can be argued (see e.g. (van der Maaten and Hinton 2008)) that this effect is desirable since it mirrors different possible views of the given data, reflecting the ill-posedness of the problem. Auxiliary information in the form of class labels can be useful to shape the problem in such settings and to resolve (parts of) the inherent ambiguities. Aspects of the data should be included into the visualization which are of particular relevance for the given class labels, while aspects can be neglected if they are not so important due to the given labeling. Thus, additional information, such as class membership information, can improve the results of dimension reduction by reducing possible "noise" in the data and keeping the essential information to discriminate the classes.

This observation has led to the development of a variety of visualization techniques which take given labels into account. These methods still map the original data to low-dimensions, but they do so using the additional information. Examples for such methods include LDA and variations, supervised NeRV (sNeRV), supervised Isomap, Multiple Relational Embedding (MRE), etc. (Venna et al. 2010), for example, give a recent overview and compare various methods for supervised data visualization. Here, we essentially repeat the experiments as proposed in (Venna et al. 2010) to demonstrate the suitability of our general method to incorporate auxiliary information into the data visualization.

In this section we show some examples of the proposed method based on the t-SNE cost function, employing supervised local linear projections $p^k(\boldsymbol{x}^i)$ (Eq. (6.19)).

Here, the parameters Ω^k and $\boldsymbol{w}^k$ are acquired by a supervised, localized prototype based classifier, LiRaM LVQ (Bunte, Hammer, Wismüller and Biehl 2010, Schneider et al. 2009a) (see Algorithm 2.3 and Algorithm 3.1 on pages 16 and 25). We compare the results to alternative state of the art techniques on the three data sets mimicking the experiments by (Venna et al. 2010):

- The *Letter recognition* data set (referred to as Letter in the following) from the UCI Machine Learning Repository (Asuncion et al. 1998). It is a 16-dimensional data set of 4×4 images of the 26 capital letters of the alphabet. These 26 classes base on 20 different distorted fonts. In total, 20000 data points are given.
- The *Phoneme* data set taken from LVQ-PAK (Kohonen et al. 1996) consists of 20-dimensional feature vectors representing phoneme samples stemming from 13 different classes.
- The *Landsat* satellite data set is contained in the UCI Machine Learning Repository. Each of the 6435 36-dimensional vectors corresponds to a 3×3 satellite image measured in four spectral bands. The six classes indicate the terrain type in the image: red soil, cotton crop, grey soil, damp grey soil, soil with vegetation stubble, and very damp grey soil.

For these data sets, we consider a projection to two dimensions by means of a locally linear function, as before, characterized by the functional form Eq. (6.21). Unlike the previous setting, this form is biased towards the given class information, because the local projections p^k are determined by means of a supervised prototype-based projection method: We used LiRaM LVQ with the rank of the localized matrices Λ^k limited to 10 (for Letter and Phoneme) and 30 (for Landsat), respectively. Based on this setting, the offsets $\boldsymbol{o}^k$ are initialized by means of the prototypes $\boldsymbol{w}^k$ centering all projections, and the projections Ω^k are given directly by the canonical representation following Eq. (3.2) of the matrices Λ^k obtained by LiRaM LVQ to get good class separation. Correspondingly, the parameter matrices A^k map from 10 or 30 dimensions to two dimensions in this case. The supervised training of the initial functional form of the mapping function, Eq. (6.21), by means of LiRaM LVQ as well as the (unsupervised) training of the free parameters of the mapping function takes place using only a small subset of the data (7%-18%) while the evaluation of the visualization takes into account the full data set.

The goal of supervised dimension reduction is the preservation of classification performance, and, is hence, quite different to classical unsupervised dimension reduction. In consequence, the quality assessment of the final embedding should be done differently. Here, following the approach of (Venna et al. 2010), we measure

the 5-nearest neighbor classification error (5NN error) of the resulting visualizations achieved in a 10-fold cross validation scheme. We compare the result obtained by locally linear projection based on the t-SNE cost function and a functional form biased by a discriminative prototype based classifier (referred to as DiReduct Map) as specified above to several state-of-the-art supervised non-linear embedding methods taken from (Venna et al. 2010):

- sNeRV (Venna et al. 2010) which uses input distances $d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{x}^j)$ induced by the Fisher information from a non-parametric supervised classifier.
- MRE (Memisevic and Hinton 2005) which is an extension of SNE accommodating additional characteristics of the data space or subspaces provided as similarity relations priorly known to the user.
- Colored MVU (cMVU) (Song et al. 2008) is an extension of the unsupervised MVU. It is also called maximum unfolding via Hilbert-Schmidt independence criterion (MUHSIC), because it maximizes the dependency between the embedding coordinates and the labels.
- supervised Isomap (S-Isomap) (Geng et al. 2005) is an extension of unsupervised Isomap extending distances to incorporate label information in an ad hoc manner.
- Parametric Embedding (PE) (Iwata et al. 2007) aims at the preservation of the topology of the original data by minimizing a sum of Kullback-Leibler divergences between a Gaussian mixture model in the original and embedding space.
- Neighborhood Component Analysis (NCA) (Goldberger et al. 2004) adapts a metric by finding a linear transformation of the original data such that the average leave-one-out k-nearest neighbor classification performance is maximized in the transformed space (see Section 3.4.2 for details).

Note that these methods constitute representative supervised visualization techniques which enrich dimensionality reduction by incorporating given label information in various forms.

The error rates of the nearest neighbor classification (using squared Euclidean distance) on the whole original high-dimensional data set and after dimension reduction with the different methods are shown in Figure 6.3. In contrast to our method, the other techniques were evaluated using only a small subset of the data sets (only 1500 sampled points), because they are based on the embedding of single points. For our approach, we train on a subsample of 7% only, but also report the

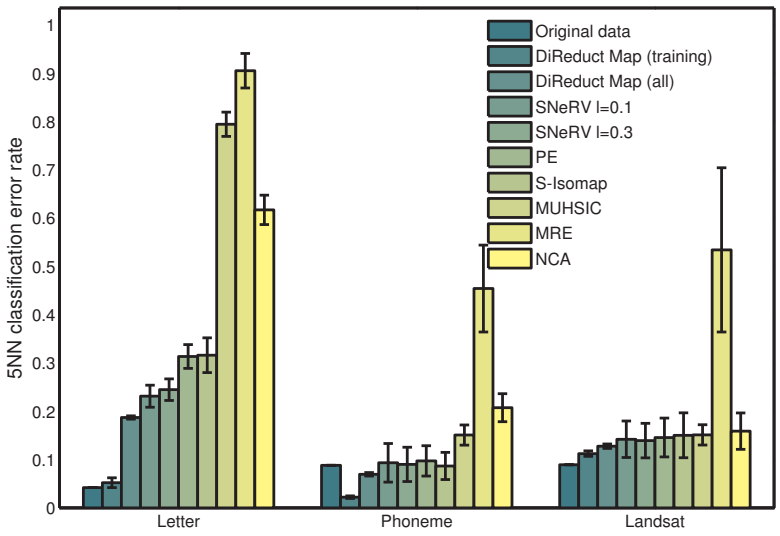


Figure 6.3: 5-nearest neighbor errors of supervised visualization on three data sets.

results of the full data set obtained by the explicit mapping. Note that the classification error obtained by an explicit mapping biased according to auxiliary information is smaller than the alternatives for all three data sets. It is remarkable, that the error in the reduced space is also comparable to the error on the high-dimensional data for most data sets. For the Phoneme data set the supervised dimension reduction even leads to a better separation of the classes than in the original space. Hence the proposed method displays excellent generalization, this way offering an efficient technique to deal with large data sets by inferring a mapping on a small subset only. Example visualizations of the proposed method are displayed in Figure 6.4. A clear class structure is visible especially for the data sets Letter and Phoneme. Interestingly, the Letter clusters arrange in a quite intuitive way: "O", "Q", "G" and "C" stay close together, so do "M", "N" and "H". The qualitative characteristic of the projections is the same for the training data and the full data sets, displaying the excellent generalization ability of the proposed method.

6.7 Generalization ability and complexity

The introduction of a general view on dimension reduction as cost optimization extends the existing techniques to large data sets by subsampling. A mapping func-

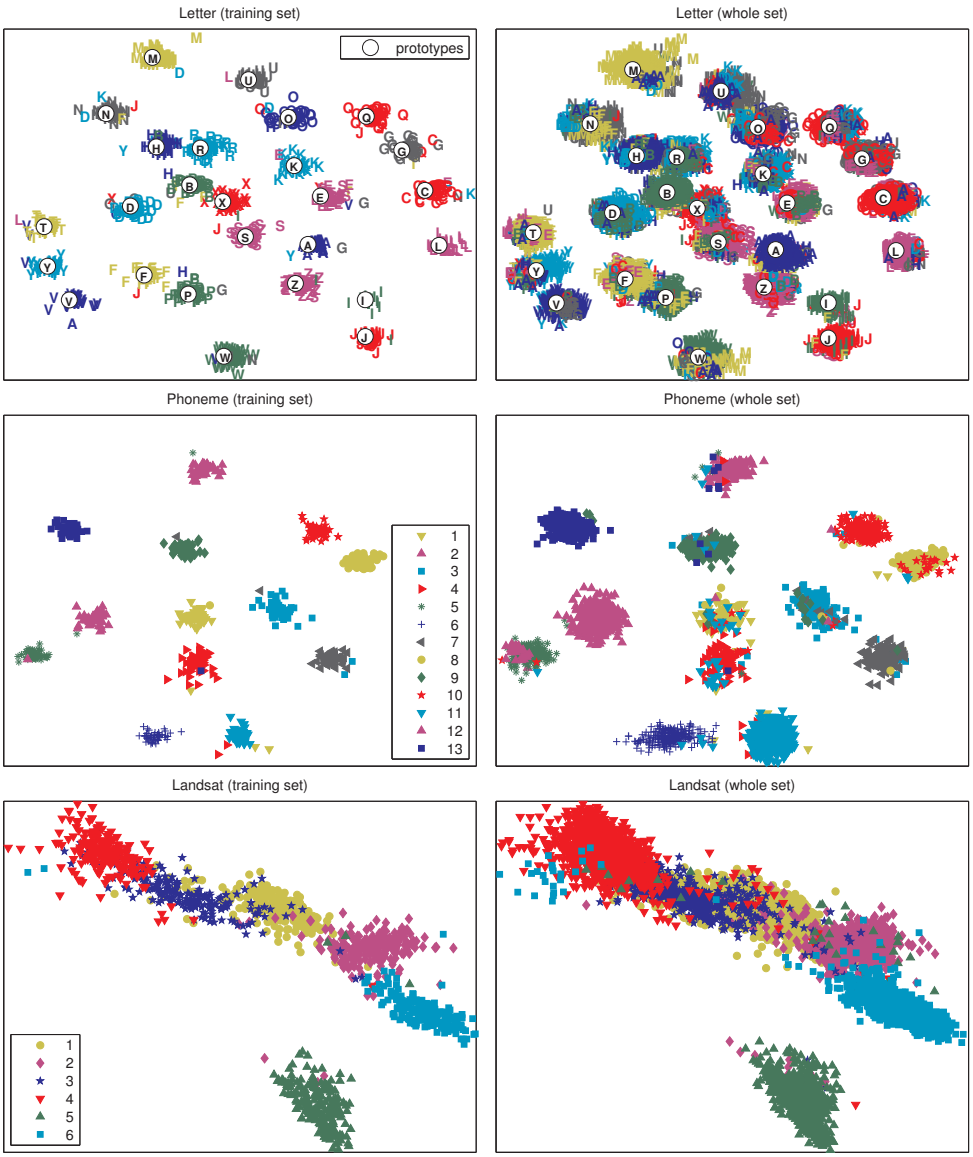


Figure 6.4: Examples of supervised visualizations of the data sets in two dimensions. The result of the subsample used for training as well as the full data set are depicted.

tion f_W based on a small data subset is obtained which extends the embedding to arbitrary points $\boldsymbol{x}$ coming from the same distribution as the training samples. In this context it is of particular interest if the procedure can be substantiated by mathematical guarantees concerning its generalization ability. We are interested in the question if a mapping achieves good quality on arbitrary data assuming it showed satisfactory embeddings on a finite subset, which has been used to determine the mapping parameters.

A formal evaluation measure of dimensionality reduction has been proposed by (Lee and Verleysen 2009, Venna et al. 2010), based on the measurement of local neighborhoods and their preservation while projecting the data. Since these measures rely on a finite number of neighbors, they are not directly suited as evaluation measures for arbitrary data distributions in $\mathbb{R}^N$. Furthermore, restrictions on the applicability of these quality measures to evaluate clusterings, have been published recently by (Mokbel et al. 2010).

6.7.1 A possible formalization

As pointed out by (Lee and Verleysen 2009) one alternative objective of dimension reduction is to preserve the available information as much as possible – this objective is usually hardly used to evaluate non-parametric dimensionality reduction because it cannot be evaluated due to the lack of an explicit mapping. Given an explicit mapping, however, it can act as a valid evaluation measure: the error of a dimensionality reduction mapping f is defined as

$$E(P) := \int_{\mathcal{X}} \|\boldsymbol{x} - f^{-1}(f(\boldsymbol{x}))\|^2 P(\boldsymbol{x})\, d\boldsymbol{x} \tag{6.26}$$

where P defines the probability measure according to which the data $\boldsymbol{x}$ are distributed in $\mathcal{X}$ and f^{-1} denotes an approximate inverse mapping of f; an exact inverse might not exist in general, but local inversion is usually possible apart from sets of measure 0. In practice the full data manifold is not available such that this objective can neither be evaluated nor optimized given a finite data set. Rather, the empirical error

$$\widehat{E}_n(\boldsymbol{x}) := \frac{1}{n} \sum_i \|\boldsymbol{x}^i - f^{-1}(f(\boldsymbol{x}^i))\|^2 \tag{6.27}$$

can be computed based on given data samples $\boldsymbol{x}^i$. A dimension reduction mapping shows good generalization ability iff the empirical error $\widehat{E}_n(\boldsymbol{x})$ is representative for the true error $E(P)$. If the form of f is fixed prior to training, we can specify a function class $\mathcal{F}$ with $f \in \mathcal{F}$ independently of the given training set. Assuming representative vectors $\boldsymbol{x}^i$ are chosen independently and identically distributed according

to P the question is whether the empirical error allows to limit the real error $E(P)$ we are interested in. As usual, bounds should hold simultaneously for all possible functions in $\mathcal{F}$ to circumvent the problem that the function f is chosen according to the given training data.

This setting can be captured in the classical framework of computational learning theory, as specified e.g. by (Bartlett and Mendelson 2003). We can adapt Theorem 8 of (Bartlett and Mendelson 2003) to our setting: We assume that the norm of the input data is limited to the unit ball. Possibly, prior normalization is necessary, which would be mirrored by corresponding constants in the bounds. We consider the loss function

$$\mathcal{L} : \mathcal{X} \times \mathcal{X} \to [0,1] \quad , \quad (\boldsymbol{x}^i, \boldsymbol{x}^j) \to \|\boldsymbol{x}^i - \boldsymbol{x}^j\|^2 \ . \tag{6.28}$$

Then, as reported by (Bartlett and Mendelson 2003) (Theorem 8), assuming i.i.d. data according to P, for any confidence $\delta \in (0,1)$ and every $f \in \mathcal{F}$ the relation

$$E(P) \leqslant \widehat{E}_n(\boldsymbol{x}) + R_n(\mathcal{L}_{\mathcal{F}}) + \sqrt{\frac{8 \ln(2/\delta)}{n}} \tag{6.29}$$

holds with probability at least $1 - \delta$ where

$$\mathcal{L}_{\mathcal{F}} := \{\boldsymbol{x} \mapsto \mathcal{L}(f^{-1}(f(\boldsymbol{x})), \boldsymbol{x}) \mid f \in \mathcal{F}\} \tag{6.30}$$

and R_n refers to the so-called Rademacher complexity of the function class. The Rademacher complexity constitutes a quantity which, similar to the Vapnik Chervonenkis dimension, estimates the capacity of a given function class, see (Bartlett and Mendelson 2003). The Rademacher complexity of many function classes (such as piecewise constant, piecewise linear functions with a fixed number of pieces, or polynomials of fixed degree) can be limited by a term which scales as $n^{-1/2}$. See (Bartlett and Mendelson 2003) for structural results and explicit bounds for e.g. linear functions, and e.g. (Schneider et al. 2009a) for explicit bounds on piecewise constant functions as induced by prototype based clustering. This result implies that the generalization ability of dimension reduction mappings as considered above can be guaranteed in principle since the Gaussian complexity of the class $\mathcal{L}_{\mathcal{F}}$ can be limited in our settings. It remains a subject of future research to find explicit good bounds.

6.7.2 Computational complexity

Assume a set $\mathbf{X}$ of points is given. Most dimensionality reduction techniques are computationally quite demanding due to the form if the overall costs Eq. (6.12): since, usually, the characteristics map sequences of points to sequences of real values

of the same length, the computation of Eq. (6.12) is at least $\mathcal{O}(|\mathbf{X}|^2)$. This is infeasible for large $\mathbf{X}$. Out-of-sample extensions by means of an implicit mapping depend on a subset $\mathbf{X}_0 \subset \mathbf{X}$ only. If the principle as derived in this paper is used, the corresponding complexity is given by $\mathcal{O}(|\mathbf{X}_0|^2 + |\mathbf{X}_0| \cdot |\mathbf{X}|)$, since only the subset $\mathbf{X}_0$ is mapped using the original method, afterwards, all remaining points are mapped by separately optimizing the costs of one $\boldsymbol{x} \in \mathbf{X}$ regarding their relation to $\mathbf{X}_0$, the latter being $\mathcal{O}(|\mathbf{X}_0|)$ for every $\boldsymbol{x}$. Thus, this approach substantially reduces the effort depending on the size of $\mathbf{X}_0$, but it does not easily allow a way to control the form of the mapping, or to integrate prior label information. By choosing an explicit functional form, the complexity is further reduced to $\mathcal{O}((|\mathbf{X}_0| \cdot |W|)^2 + |W| \cdot |\mathbf{X}|)$, assuming an effort $\mathcal{O}(|W|)$ to evaluate f_W. Since, usually, $|\mathbf{X}| \gg |\mathbf{X}_0| \gg |W|$, this constitutes a further considerable reduction of the time required to map all points.

6.8 Conclusion

In this contribution we reformulated dimension reduction as an optimization problem based on structural characteristics. As a consequence many popular nonparametric dimension reduction techniques can simultaneously be extended to learn an explicit mapping function. The optimization of a parametrized mapping function for dimension reduction is beneficial in several ways: large data sets can be dealt with because the mapping function can be learned on a small random subset of the data. Furthermore this framework allows us to consider the generalization ability of dimension reduction since an explicit cost function is available in terms of the reconstruction error. Interestingly, bounds as derived in the context of computational learning theory can be transferred to this setting.

We showed the suitability of the approach based on the integration of global linear and locally linear projections into the t-SNE dimension reduction method on different data sets. Furthermore we show the integration of auxiliary (e.g. class) information into the framework. The proposed general framework is very flexible and can be combined with every possible form of the mapping function. The investigation of alternative dimension reduction mappings based on other cost functions and other functional forms of the mapping, as well as the derivation of explicit bounds on its generalization ability will be the subject of future work. At present, the settings have been restricted to Euclidean data only due to the form of the mapping f_W. Naturally, more general forms could be considered which can take more complex, non-Euclidean data as inputs, such as mappings which are based on general dissimilarity characterization. Since it is not possible to embed such data in any Euclidean vector space, possibly qualitatively different results may occur.

6.A Derivatives for dimension reduction mappings

6.A.1 Derivatives of the linear t-SNE mapping

Here we show the derivatives of the t-SNE cost function Eq. (6.8) assuming a linear mapping function f_W of the high-dimensional data points $\boldsymbol{x}$, see Eq. (6.17). We use the following abbreviations

$$q_{ij} = \frac{(1 + d_{\mathcal{E}}(\widehat{\boldsymbol{\xi}}^i, \widehat{\boldsymbol{\xi}}^j)/\varsigma)^{-\frac{\varsigma+1}{2}}}{\sum_{k\neq l}(1 + d_{\mathcal{E}}(\widehat{\boldsymbol{\xi}}^k, \widehat{\boldsymbol{\xi}}^l)/\varsigma)^{-\frac{\varsigma+1}{2}}} = \frac{(1 + \frac{d_{ij}}{\varsigma})^{-\frac{\varsigma+1}{2}}}{\sum_{k\neq l}(1 + \frac{d_{kl}}{\varsigma})^{-\frac{\varsigma+1}{2}}} = \frac{g_\varsigma^{ij}}{\sum_{k\neq l} g_\varsigma^{kl}} \tag{6.31}$$

in the context of the probabilities of neighborhoods in the low-dimensional space. The rectangular matrix A defines the linear mapping from $\mathbb{R}^N \to \mathbb{R}^M$. This matrix may be optimized using a stochastic gradient descent procedure using following gradient:

$$\begin{aligned}
\frac{\partial E_{\mathrm{t-SNE}}}{\partial A} &= \sum_i \sum_j \frac{\partial E_{\mathrm{t-SNE}}}{\partial q_{ij}} \cdot \frac{\partial q_{ij}}{\partial d_{\mathcal{E}}(\widehat{\boldsymbol{\xi}}^i, \widehat{\boldsymbol{\xi}}^j)} \cdot \frac{\partial d_{\mathcal{E}}(\widehat{\boldsymbol{\xi}}^i, \widehat{\boldsymbol{\xi}}^j)}{\partial A} \\
&= \sum_i \sum_j -p_{ij} \cdot \frac{\partial \log(q_{ij})}{\partial A} \quad = \sum_i \sum_j -\frac{p_{ij}}{q_{ij}} \cdot \frac{\partial q_{ij}}{\partial A} \\
&= \sum_i \sum_j -\frac{p_{ij}}{q_{ij}} \cdot \frac{\frac{\partial g_\varsigma^{ij}}{\partial A} \cdot \sum_{k\neq l} g_\varsigma^{kl} - g_\varsigma^{ij} \cdot \sum_{k\neq l} \frac{\partial g_\varsigma^{kl}}{\partial A}}{\left(\sum_{k\neq l} g_\varsigma^{kl}\right)^2} \\
&= \sum_i \sum_j -\frac{p_{ij} \cdot \left(-\frac{\varsigma+1}{2\varsigma}\right)}{q_{ij}} \cdot \\
&\quad \frac{g_\varsigma^{ij} \cdot \left(1 + \frac{d_{ij}}{\varsigma}\right)^{-1} \frac{\partial d_{ij}}{\partial A} \cdot \sum_{k\neq l} g_\varsigma^{kl} - g_\varsigma^{ij} \cdot \sum_{k\neq l} g_\varsigma^{kl} \cdot \left(1 + \frac{d_{kl}}{\varsigma}\right)^{-1} \frac{\partial d_{kl}}{\partial A}}{\left(\sum_{k\neq l} g_\varsigma^{kl}\right)^2} \\
&= \sum_i \sum_j -\frac{p_{ij} \cdot \left(-\frac{\varsigma+1}{2\varsigma}\right)}{q_{ij}} \cdot \\
&\quad \left[q_{ij} \left(1 + \frac{d_{ij}}{\varsigma}\right)^{-1} \frac{\partial d_{ij}}{\partial A} - q_{ij} \cdot \sum_{k\neq l} q_{kl} \cdot \left(1 + \frac{d_{kl}}{\varsigma}\right)^{-1} \frac{\partial d_{kl}}{\partial A} \right] \\
&= \frac{\varsigma+1}{2\varsigma} \sum_{ij} p_{ij} \left[\left(1 + \frac{d_{ij}}{\varsigma}\right)^{-1} \frac{\partial d_{ij}}{\partial A} - \sum_{k\neq l} q_{kl} \left(1 + \frac{d_{kl}}{\varsigma}\right)^{-1} \frac{\partial d_{kl}}{\partial A} \right] \\
&= \frac{\varsigma+1}{2\varsigma} \sum_{i\neq j} (p_{ij} - q_{ij}) \cdot \left(1 + \frac{d_{ij}}{\varsigma}\right)^{-1} \frac{\partial d_{ij}}{\partial A}
\end{aligned}$$

With Euclidean distance $d_{ij} = d_{\mathcal{E}}(\widehat{\boldsymbol{\xi}}^i, \widehat{\boldsymbol{\xi}}^j) = ||A\boldsymbol{x}^i - A\boldsymbol{x}^j||^2$ follows:

$$\frac{\partial d_{ij}}{\partial A} = 2(A\boldsymbol{x}^i - A\boldsymbol{x}^j)(\boldsymbol{x}^i - \boldsymbol{x}^j) \tag{6.32}$$

$$\frac{\partial E_{\mathrm{t-SNE}}}{\partial A} = \frac{\varsigma + 1}{\varsigma} \sum_i \sum_j \frac{(p_{ij} - q_{ji})}{1 + \frac{1}{\varsigma}||A\boldsymbol{x}^i - A\boldsymbol{x}^j||^2} (A\boldsymbol{x}^i - A\boldsymbol{x}^j)(\boldsymbol{x}^i - \boldsymbol{x}^j) \ . \tag{6.33}$$

6.A.2 Derivatives of local linear t-SNE mappings

The derivatives of the t-SNE cost function using a local linear mapping function following Eq. (6.21) based on the linear projections Eq. (6.19) can be achieved in analogy to above:

$$\begin{aligned}\frac{\partial E_{\mathrm{t-SNE}}}{\partial \boldsymbol{o}^k} &= \sum_{ij} \frac{\partial E_{\mathrm{t-SNE}}}{\partial q_{ij}} \cdot \frac{\partial q_{ij}}{\partial d_{\mathcal{E}}(\widehat{\boldsymbol{\xi}}^i, \widehat{\boldsymbol{\xi}}^j)^2} \cdot \frac{\partial d_{\mathcal{E}}(\widehat{\boldsymbol{\xi}}^i, \widehat{\boldsymbol{\xi}}^j)^2}{\partial \boldsymbol{o}^k} \\ &= \frac{\varsigma + 1}{\varsigma} \sum_{ij} \frac{(p_{ij} - q_{ji})}{1 + \frac{1}{\varsigma} d_{\mathcal{E}}(\widehat{\boldsymbol{\xi}}^i, \widehat{\boldsymbol{\xi}}^j)^2} \cdot (\widehat{\boldsymbol{\xi}}^i - \widehat{\boldsymbol{\xi}}^j)(r_{ik} - r_{jk})\end{aligned} \tag{6.34}$$

and

$$\begin{aligned}\frac{\partial E_{\mathrm{t-SNE}}}{\partial A^k} &= \sum_{ij} \frac{\partial E_{\mathrm{t-SNE}}}{\partial q_{ij}} \cdot \frac{\partial q_{ij}}{\partial d_{\mathcal{E}}(\widehat{\boldsymbol{\xi}}^i, \widehat{\boldsymbol{\xi}}^j)^2} \cdot \frac{\partial d_{\mathcal{E}}(\widehat{\boldsymbol{\xi}}^i, \widehat{\boldsymbol{\xi}}^j)^2}{\partial A^k} \\ &= \frac{\varsigma + 1}{2\varsigma} \sum_{ij} \frac{(p_{ij} - q_{ji})}{1 + \frac{1}{\varsigma} d_{\mathcal{E}}(\widehat{\boldsymbol{\xi}}^i, \widehat{\boldsymbol{\xi}}^j)^2} \cdot \frac{\partial d_{\mathcal{E}}(\widehat{\boldsymbol{\xi}}^i, \widehat{\boldsymbol{\xi}}^j)^2}{\partial A^k} \\ &= \frac{\varsigma + 1}{\varsigma} \sum_{ij} \frac{(p_{ij} - q_{ji})}{1 + \frac{1}{\varsigma} d_{\mathcal{E}}(\widehat{\boldsymbol{\xi}}^i, \widehat{\boldsymbol{\xi}}^j)^2} (\widehat{\boldsymbol{\xi}}^i - \widehat{\boldsymbol{\xi}}^j)(r_{ik} p^k(\boldsymbol{x}^i) - r_{jk} p^k(\boldsymbol{x}^j)) \ .\end{aligned} \tag{6.35}$$

Published as:

K. Bunte, B. Hammer, A. Wismüller and M. Biehl – *"Adaptive Local Dissimilarity Measures for Discriminative Dimension Reduction of Labeled Data,"* Neurocomputing, vol. 73(7-9), pp. 1074–1092, 2010.

K. Bunte, B. Hammer and M. Biehl – *"Nonlinear Dimension Reduction and Visualization of Labeled Data,"* in International Conference of Computer Analysis of Images and Patterns (CAIP), pp. 1162–1170, 2009.

K. Bunte, B. Hammer, P. Schneider and M. Biehl – *"Nonlinear Discriminative Data Visualizations,"* in Proc. of 17th European Symposium on Artificial Neural Networks (ESANN), pp. 65–70, Belgium, 2009.

Chapter 7

Adaptive Local Dissimilarity Measures for Discriminative Dimension Reduction and Visualization

You learn more quickly under the guidance of experienced teachers. You waste a lot of time going down blind alleys if you have no one to lead you.

W. Somerset Maugham (1874 - 1965)

Abstract

Since embedding in lower dimensions necessarily includes a loss of information, methods to explicitly control the information kept by a specific dimensionality reduction technique are highly desirable. The incorporation of class information constitutes an important specific case. The aim is to preserve and potentially enhance the discrimination of classes in lower dimensions. In this chapter we use the extension of prototype-based local distance learning introduced in Part I of this thesis, which results in a discriminative dissimilarity measure for a given labeled data manifold. The adapted local distance measure can be used as basis for unsupervised dimensionality reduction techniques, which take into account neighborhood information. We show the combination of different dimensionality reduction techniques with a discriminative similarity measure learned by LiRaM LVQ using local projections Ω^j and their behavior with different parameter settings. The methods are discussed in terms of artificial and real world data sets.

7.1 Introduction

In the last decades an enormous number of unsupervised dimensionality reduction methods has been proposed. In general, this constitutes an ill-posed prob-

lem since a clear specification which properties of the data should be preserved, is missing. Standard criteria, for instance the distance measure employed for neighborhood assignment, may turn out unsuitable for a given data set, and relevant information often depends on the situation at hand. If data labeling is available, the aim of dimensionality reduction can be defined clearly: the preservation of the classification accuracy in a reduced feature space. Supervised linear dimension reducers are for example the LiRaM LVQ (Bunte, Schneider, Hammer, Schleif, Villmann and Biehl 2011) introduced in Section 3.2, LDA (Fukunaga 1990), Targeted Projection Pursuit (TPP) (Faith et al. 2006), and discriminative component analysis (Peltonen et al. 2006). Often, however, the classes cannot be separated by a linear classifier while a non-linear data projection better preserves the relevant information. Examples for nonlinear discriminative visualization techniques include, extensions of the SOM incorporating class labels (Villmann et al. 2006) or more general auxiliary information (Peltonen et al. 2004). In both cases, the metric of SOM is adjusted such that it emphasizes the given auxiliary information and, consequently, SOM displays the aspects relevant for the given labeling. Further supervised dimensionality reduction techniques are model-based visualization (Kontkanen et al. 2000), sNeRV (Venna et al. 2010), MRE (Memisevic and Hinton 2005), cMVU (Song et al. 2008), S-Isomap (Geng et al. 2005), PE (Iwata et al. 2007) and NCA (Goldberger et al. 2004), already mentioned in Section 3.4.2 and Section 6.6. In addition, linear schemes such as LDA can be kernelized yielding a nonlinear supervised dimensionality reduction scheme (Baudat and Anouar 2000). These models have the drawback that they are often very costly (squared or cubic with respect to the number of data points). Recent approaches provide scalable alternatives, sometimes at the cost of non convexity of the problem (Kulis et al. 2007, Vasiloglou et al. 2008, Collobert et al. 2006). However, in most methods, the kernel has to be chosen prior to training and no metric adaptation according to the given label information takes place.

Here, we aim in the identification and investigation of principled possibilities to combine an adaptive metric and recent visualization techniques towards a discriminative approach. We will exploit the discriminative scheme exemplary for different types of visualization, necessarily restricting the number of possible combinations to exemplary cases. A number of alternative combinations of metric learning and data visualization as well as principled alternatives to arrive at discriminative visualization techniques (such as e.g. colored Maximum Variance Unfolding (Song et al. 2008)) were addressed for example in Section 6.6. In this Chapter we combine prototype-based matrix learning schemes, which result in local discriminative dissimilarity measures and local linear projections of the data, with different neighborhood based nonlinear dimensionality reduction techniques and a charting technique. In a first step the dissimilarity measure is learned using the LGMLVQ

(see Algorithm 2.4) based on localized matrices Λ^j, possibly limiting the rank as proposed in Chapter 3. In the second step unsupervised techniques like manifold charting (Brand 2002), Isomap (Tenenbaum et al. 2000), LLE (Roweis and Saul 2000), the Exploration Observation Machine (XOM) (Wismüller 2009d) and SNE (Hinton and Roweis 2003) are performed incorporating the supervised information from the Learning Vector Quantization (LVQ) approach. This leads to supervised nonlinear dimensionality reduction and visualization techniques.

The following section gives a short overview over the techniques. We focus on the question in how far local linear discriminative data transformations as provided by LGMLVQ and LiRaM LVQ offer principled possibilities to extend standard unsupervised visualization tools to discriminative visualization. Section 7.3 discusses the different approaches for one artificial and three real world data sets and compares the results to popular supervised as well as unsupervised dimensionality reduction techniques. Finally we conclude in section 7.4.

7.2 Supervised Nonlinear Dimension Reduction

For general data sets a global linear reduction to lower dimensions may not be sufficient to preserve the information relevant for classification. In (van der Maaten et al. 2009) it is argued that the combination of several local linear projections to a nonlinear mapping can yield promising results. We use this concept and learn discriminative local linear (probably low-dimensional) projections from labeled data using an efficient prototype based learning scheme, LGMLVQ (see Algorithm 2.4) possibly limiting the rank using local transformations $\Omega^j \in \mathbb{R}^{M\times N}$ with $M \leqslant N$ (following the principle of LiRaM LVQ Section 3.2). Locally linear projections obtained from this first step provide transformations of the data points, which aims in the preservation of the information relevant for the classification. Instead of the local coordinates, local distances d^{Λ^j} Eq. (2.22) induced by these local representation of data can be considered. As a consequence, visualization techniques which rely on local coordinate systems or distances, respectively, can be combined with this adaptive dissimilarity to arrive at a discriminative global nonlinear projection method. This way, an incorporation into techniques such as manifold charting (Brand 2002), Isomap (Tenenbaum et al. 2000), LLE (Roweis and Saul 2000), SNE (Hinton and Roweis 2003), and the XOM (Wismüller 2009d), among others becomes possible.

7.2.1 LiRaM LVQ for discriminative visualization

In contrast to Localized LiRaM LVQ (LLiRaM LVQ) (see Section 3.2.1) we do not consider an adaptive matrix composed of two matrices $\Lambda^j = \Omega^\top \Psi^{j\top} \Psi^j \Omega$ in this Chap-

ter. Instead we adopt the localized dissimilarity measure d^{Λ^j} Eq. (2.22) of LGMLVQ (summarized in Algorithm 2.4) assuming a possible limit of the rank of $\Lambda^j = \Omega^{j\top}\Omega$ by $\Omega^j \in \mathbb{R}^{M\times N}$ with $M \leqslant N$. As for the GMLVQ method the LGMLVQ algorithm and its derivatives do not change in case of a limited rank. Because of the kinship we refer to this adaptation of LGMLVQ also as LiRaM LVQ, having in mind that we do not address global linear projections Ω, but local linear Ω^j in this Chapter.

For every prototype, a low-dimensional embedding $\boldsymbol{\xi}^i$ of each data point $\boldsymbol{x}^i$, akin to Eq. (6.19) is given by:

$$p^k(\boldsymbol{x}^i) = \Omega^k \boldsymbol{x}^i = \boldsymbol{\xi}^i \quad . \tag{7.1}$$

This projection is a meaningful discriminative projection in the neighborhood of a prototype. For a data point $\boldsymbol{x}^i$ usually the projection Ω^J of its closest prototype $\boldsymbol{w}^J$ is considered. This way, a naive mapping is given as

$$\boldsymbol{x}^i \mapsto p^J(\boldsymbol{x}^i) = \Omega^J \boldsymbol{x}^i \text{ with } d^{\Lambda^J}(\boldsymbol{x}^i, \boldsymbol{w}^J) = \min_k d^{\Lambda^k}(\boldsymbol{x}^i, \boldsymbol{w}^k) \quad . \tag{7.2}$$

We will address this local linear mapping rule in the following as LiRaM LVQ projection. However, the cost function Eq. (2.23) together with the distance definition Eq. (2.22) does not ensure that these local projections align correctly and that they do not overlap when shown in one coordinate system. Rather, the projections provide widely unconnected mappings to low dimensions which offer only a locally valid visualization. Nevertheless the mapping defined by Eq. (7.2) can give a first intuition about the problematic samples and distinguish "easy" classes from more difficult ones. Therefore, we will use this projection for comparison in the experiments.

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

Local coordinates

As already stated, LiRaM LVQ gives rise to local linear projection maps p^k as defined in Eq. (7.1), which assign local projection coordinates to every data point $\boldsymbol{x}^i$. These projections can be accompanied by values which indicate the responsibility r_{ik} of mapping k for data point i. Crisp responsibilities are obtained by means of the receptive fields, setting r_{ik} to 1 iff $\boldsymbol{w}^k$ is the winner for $\boldsymbol{x}^i$ and 0 otherwise (see

Eq. (6.20)). Alternatively, soft assignments can be obtained by centering Gaussian curves of appropriate bandwidth at the prototypes and successive normalization, such that $\sum_k r_{ik} = 1$.

These two ingredients constitute a sufficient input for data visualization methods which rely on local linear projections of the data only, such as manifold charting, LLC (van der Maaten et al. 2009) and Local Tangent Space Alignment (LTSA) (Zhang and Zha 2002). Basically, those methods arrive at a global embedding of data based on local coordinates by gluing the points together such that the overall mapping is consistent with the original data points as much as possible. The methods differ in the precise cost function which is optimized: Manifold charting relying on the sum squared error of points at overlapping pieces of the local charts, while LLC focuses on the local topology and tries to minimize the reconstruction error of points from their neighborhood. Both approaches provide explicit maps of the data manifold to low dimensions, such that out-of-Sample extensions are immediate. As an example for this principle, we will investigate the combination of local linear maps and manifold charting in Section 7.2.2.

Global distances

The LVQ-based learning procedure provides discriminative local distances induced by the matrices Λ^j in the receptive field of prototype $\boldsymbol{w}^j$. In contrast to the charting approach, the ranks of the distance matrices Λ^j can be chosen larger than the embedding dimension M in these cases, using e.g. full ranks and therefore the original LGMLVQ formulation or the intrinsic dimension of the data manifold. We use the resulting parameters to define a discriminative dissimilarity measure for the given data points. We define the dissimilarity of a point $\boldsymbol{x}^i$ to a point $\boldsymbol{x}$:

$$d(\boldsymbol{x}^i, \boldsymbol{x}) = (\boldsymbol{x}^i - \boldsymbol{x})^\top \Lambda^J (\boldsymbol{x}^i - \boldsymbol{x}) \text{ where } d^{\Lambda^J}(\boldsymbol{x}^i, \boldsymbol{w}^J) = \min_k d^{\Lambda^k}(\boldsymbol{x}^i, \boldsymbol{w}^k) \qquad (7.3)$$

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

The global dissimilarities defined by Eq. (7.3) can be used directly within visualization schemes which are based on distance preservation. If necessary, the dissimilarity matrix can be symmetrized prior to the mapping. Distance based visualization

methods include classical MDS, Sammon's map, SNE, t-SNE, and the XOM, to name just a few (van der Maaten et al. 2009, Hinton and Roweis 2003, van der Maaten and Hinton 2008, Wismüller 2009d). It can be expected that the combination of the global discriminative dissimilarities as given by Eq. (7.3) yields to an appropriate visualization of the data only if the visualization method mainly focuses on the close points, since the dissimilarity of far away points can only be seen as a guess in this case. Thus, classical MDS is likely to fail, while SNE or XOM seem more promising due to their focus on local topologies. As an example, we will investigate the combination of the global dissimilarity matrix with SNE and XOM, respectively, in the following.

Local distances or neighborhood

The problem that the dissimilarity measure as defined in Eq. (7.3) should preferably only be used to compare data within a receptive field or in neighbored receptive fields is avoided by visualization techniques which explicitly rely on local distances only. Instances of such visualization techniques are given by Isomap, Laplacian Eigenmaps, LLE (van der Maaten et al. 2009) and MVU (Weinberger and Saul 2006), explained in Section 6.2. These methods use the local neighborhood of a data point, i.e. its k-NN or the points in an ϵ-ball (ϵ-neighborhood), and aim at the preservation of properties of these neighborhoods. Obviously, local neighborhoods can readily be computed based on the dissimilarities given by Eq. (7.3), thus a discriminative extensions of these methods is offered this way.

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

Now we introduce four exemplary discriminative projection methods, covering the different possibilities to combine the information given by LiRaM LVQ and diverse visualization techniques. We will compare these methods to a naive embedding directly given by the local linear maps as a baseline, LDA (Fukunaga 1990) (if applicable) as a classical linear discriminative visualization tool, and t-SNE as one of the currently most powerful unsupervised visualization techniques. Further, we will emphasize the effect of discriminative information by presenting the result of

the corresponding unsupervised projection method.

7.2.2 Combination of Local Linear Patches by Charting

The charting technique introduced in (Brand 2002) provides a frame for unsupervised dimension reduction by decomposing the sample data into locally linear patches and combining them into a single low-dimensional coordinate system. This procedure can be turned into a discriminative visualization scheme by using the low-dimensional local linear projections $p^j(\boldsymbol{x}^i) \in \mathbb{R}^M$ for every data point $\boldsymbol{x}^i$ and every local projection Ω^j provided by localized LiRaM LVQ. Afterwards, the charting method can directly be used to combine these locally linear patches: The local projections $p^j(\boldsymbol{x}^i)$ are weighted by their responsibilities r_{ij} which quantify the overlap of neighbored charts. Here we choose responsibilities induced by Gaussians centered at the prototypes, since a certain degree of overlap is needed for a meaningful charting step:

$$r_{ij} \propto \exp(-(\boldsymbol{x}^i - \boldsymbol{w}^j)^\top \Lambda^j (\boldsymbol{x}^i - \boldsymbol{w}^j)/\sigma_j) \ , \tag{7.4}$$

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

$$\sigma_j = \frac{\alpha}{k} \cdot \sqrt{\sum_{\boldsymbol{w}^l \in \mathcal{N}_k(\boldsymbol{w}^j)} d^{\Lambda_j}(\boldsymbol{w}^j, \boldsymbol{w}^l)} \ , \tag{7.5}$$

where $\mathcal{N}_k(\boldsymbol{w}^j)$ denotes the k closest prototypes of $\boldsymbol{w}^j$.

Manifold charting minimizes a convex cost function that measures the amount of disagreement between the linear models on the global coordinates of the data points. The charting technique finds affine mappings A^j from the data representations $p^j(\boldsymbol{x}^i)$ to the global coordinates that solves a weighted least-squares problem:

$$[A^1, \ldots, A^{n_w}] = \arg\min_{A^j, A^k} \sum_{i=1}^{n} r_{ij} r_{ik} \| A^j(p^j(\boldsymbol{x}^i)) - A^k(p^k(\boldsymbol{x}^i)) \|^2 \ . \tag{7.6}$$

This function is based on the idea that whenever two linear models possess a high responsibility for a data point, the models should agree on the final coordinates of that point. The cost function is formulated as the squared error corresponding to a sum of all patch-to-anchor and patch-to-patch inconsistencies and can be rewritten

as a generalized eigenvalue problem. An analytical solution can be found in closed form. The final projection is given by the mapping

$$\boldsymbol{x}^i \mapsto \boldsymbol{\xi}^i = \sum_j r_{ij} \cdot A^j(p^j(\boldsymbol{x}^i)) \ . \tag{7.7}$$

We refer to (Brand 2002) for further details. Interestingly, an explicit map of the data manifold to low dimensions is obtained this way. Further, the charting step is linear in the number of data points n. We refer to the extension of charting by local discriminative projections as charting$^+$ in the following.

7.2.3 Discriminative Locally Linear Embedding

LLE (Roweis and Saul 2000) aims in the preservation of topologies induced by local k-ary neighborhoods. The idea is to reconstruct each point $\boldsymbol{x}^i$ by a linear combination of its nearest neighbors and to project data points into lower dimensions, such that this local representation of the data is preserved as much as possible. The method is summarized in Algorithm 6.1 in Section 6.2. Step 1 of the LLE algorithm is the determination of neighbors $\mathcal{N}_i$ for each data point $\boldsymbol{x}^i$. Following the ideas of supervised LLE (Wang et al. 2006) and probability-based LLE (Zhao and Zhang 2009) we take the label information into account by using the distance measure defined in Eq. (7.3) to determine the k-NNs of each point. The rest of the LLE approach remain unchanged. We refer to this discriminative extension of LLE by LLE$^+$ in the following.

7.2.4 Discriminative Isomap

Isomap (Tenenbaum et al. 2000) is an extension of metric MDS using graph distances as an approximation of the geodesic distances in the high-dimensional space. For this purpose, a weighted neighborhood graph is constructed by connecting points i and j if their distance is smaller than ϵ (ϵ-Isomap), or if j is one of the k-NNs of i (k-Isomap). Global distances between points are computed using shortest paths in this neighborhood graph, see also Section 6.2. The local neighborhood graph can serve as an interface to incorporate discriminative information provided by LiRaM LVQ. We use the distances defined by Eq. (7.3) to determine the k-NNs and to weight the edges in the neighborhood graph. Afterwards, we simply apply the same projection technique as original Isomap. We refer to this discriminative extension of Isomap as Isomap$^+$ in the following.

7.2.5 Discriminative Stochastic Neighbor Embedding

SNE constitutes an unsupervised projection which follows a probabilistic approach. It aims in the preservation of local topologies induced by the probability densities in the original space $p_{j|i}$ and the projection space $q_{j|i}$, see Section 6.2. SNE tries to find a low-dimensional data representation that minimizes the mismatch between those distributions. This is done by the minimization of the sum of the Kullback-Leibler divergences Eq. (6.6). It is easily possible to incorporate discriminative information into SNE by choosing the distances $d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{x}^j)$ in Eq. (6.4) as discriminative distances as provided by Eq. (7.3). Then, the subsequent steps can be done in the same way as in standard SNE.

7.2.6 Discriminative Exploration Observation Machine (XOM)

XOM has recently been introduced as a novel computational framework for structure-preserving dimension reduction (Wismüller 2009c, Wismüller 2009a). It can be seen as an extension of the SOM changing the interpretation of the variables. The XOM aims in the preservation of a topology in the high-dimensional space denoted by neighborhood couplings $d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{x}^j)$ between the input data points $\boldsymbol{x}$, represented by a so-called cooperativity (or neighborhood) function, e.g. a Gaussian. The low-dimensional counterparts $\boldsymbol{\xi}$ of every data point are moved in the low-dimensional space according to this neighborhood function, such that local neighborhoods are preserved. This algorithm will be explained in more detail in Chapter 8, where it is also extended further. Obviously, discriminative information can be included into XOM by substituting the distances $d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{x}^j)$ by the discriminative distances as provided by Eq. 7.3.

7.2.7 Discriminative Maximum Variance Unfolding (MVU)

MVU (Weinberger and Saul 2006) is based on a neighborhood graph with k nearest neighborhood graphs or ϵ-neighborhoods $\mathcal{N}$. Projections $\boldsymbol{\xi}^i$ are determined by maximizing the variance of the projection. The aim is, that neighboring points $\boldsymbol{x}^i$ and $\boldsymbol{x}^j$ preserve their affinities also in the low-dimensional space after projection: $d_{\mathcal{E}}(\boldsymbol{\xi}^i, \boldsymbol{\xi}^j) = d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{x}^j)$. The method is summarized in Algorithm 6.3 in Section 6.2. To include supervision in this dimension reduction technique the distance defined by Eq. (7.3) can be used to determine the k nearest neighbors. Afterwards we simply apply the same optimization as original MVU. For our experiments we used the library for semi-definite programming called CSDP[1] and the MVU implementation

[1] http://infohost.nmt.edu/~borchers/csdp.html

provided by Kilian Q. Weinberger[2].

7.2.8 Further embedding techniques

We will compare the results obtained within this discriminative framework to a few standard embedding techniques. More precisely, we will display the results of LDA (Fukunaga 1990) as a classical linear discriminative projection technology, t-SNE as an extension of SNE which constitutes one of the most promising unsupervised projection techniques available today.

LDA constitutes a supervised projection and classification technique. Given data points and corresponding labeling, it determines a global linear map such that the distances within classes of projected points are minimized whereas the distances between classes of projected points are maximized. This objective can be formalized in such a way that an explicit analytical solution is obtained by means of eigenvalue techniques. It can be shown that the maximum dimension of the projection has to be limited to $C - 1$, C being the number of classes, to give meaningful results. Hence, this method can only be applied for data sets with 3 or more classes. Further, the method is restricted to linear maps and it relies on the assumption that classes can be represented by unimodal clusters, which can lead to severe limitations in practical applications. t-SNE constitutes an extension of SNE, which achieved very promising visualization for a couple of benchmarks (van der Maaten and Hinton 2008). The probability densities in the low-dimensional space q_{ij} are defined using a student-t distribution instead of a Gaussian, see Eq. (6.10). Further details can be found in Section 6.2.

7.3 Experiments

7.3.1 Three Tip Star

This artificial dataset consists of 3000 samples in $\mathbb{R}^{10}$ with two overlapping classes (C1 and C2), each forming three clusters as displayed in Fig. 7.1. The first two dimensions contain the information whereas the remaining eight dimensions contribute high variance noise. Following the advise "always try Principal Component Analysis (PCA) first"[3] we achieve a leave-one-out 1-NN error of 29% in the data set mapped to two dimensions (the result is shown in Fig. 7.2 left panel). The best embedding of t-SNE was achieved by setting the perplexity to 35 and is shown in the right panel. The localized LiRaM LVQ was trained for $t_{\max} = 500$ epochs, with three

[2] http://www.weinbergerweb.net/Downloads/MVU.html
[3] John A. Lee, private communication, 2009.

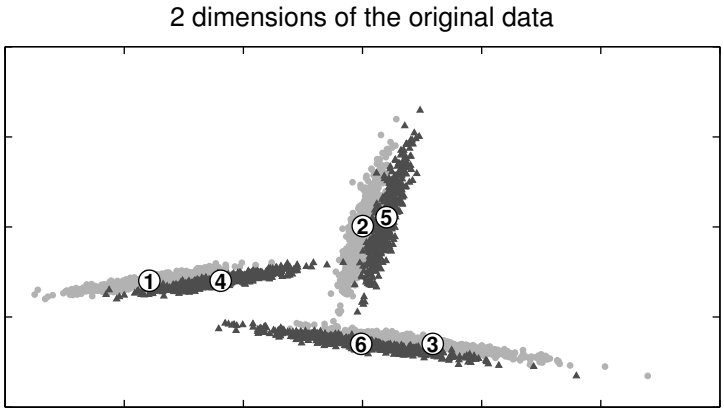


Figure 7.1: The two informational dimensions of the original Three Tip Star data set.

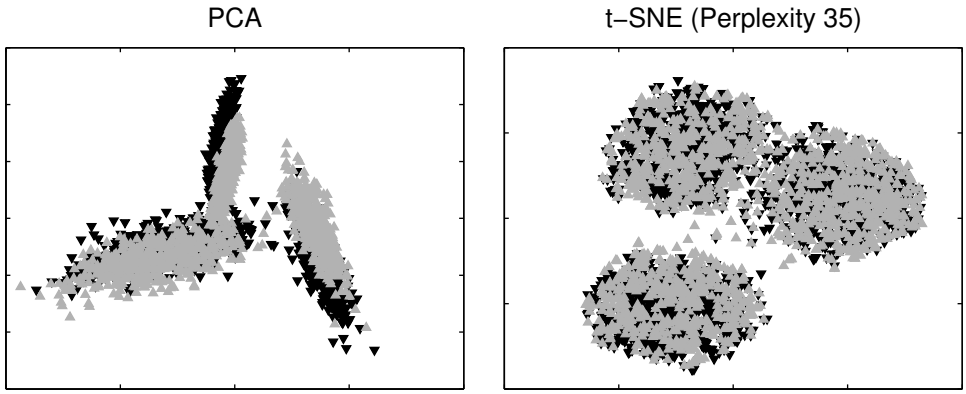


Figure 7.2: Example Visualizations of the Three Tip Star data set.

prototypes per class and local matrices of target dimension $M = 2$. Each of the prototypes was initialized close to one of the cluster centers. Initial elements of Ω^j were generated randomly according to a uniform distribution in $[-1, 1]$ with subsequent normalization of the matrix following Eq. (2.21). The learning rate for prototype vectors follows the schedule Eq. (3.5) $\tau_1(t) = 0.01/(1+(t_{\max}-1)\cdot 0.001)$ and metric learning starts at epoch $t_M = 50$ with learning rate $\tau_2(t) = 0.001/(1 + (t_{\max} - 50) \cdot 0.0001)$. We repeat localized LiRaM LVQ with 10 independent random initializations of the prototypes and matrices. The resulting mean classification error on the Three Tip Star data set is 9.7%.

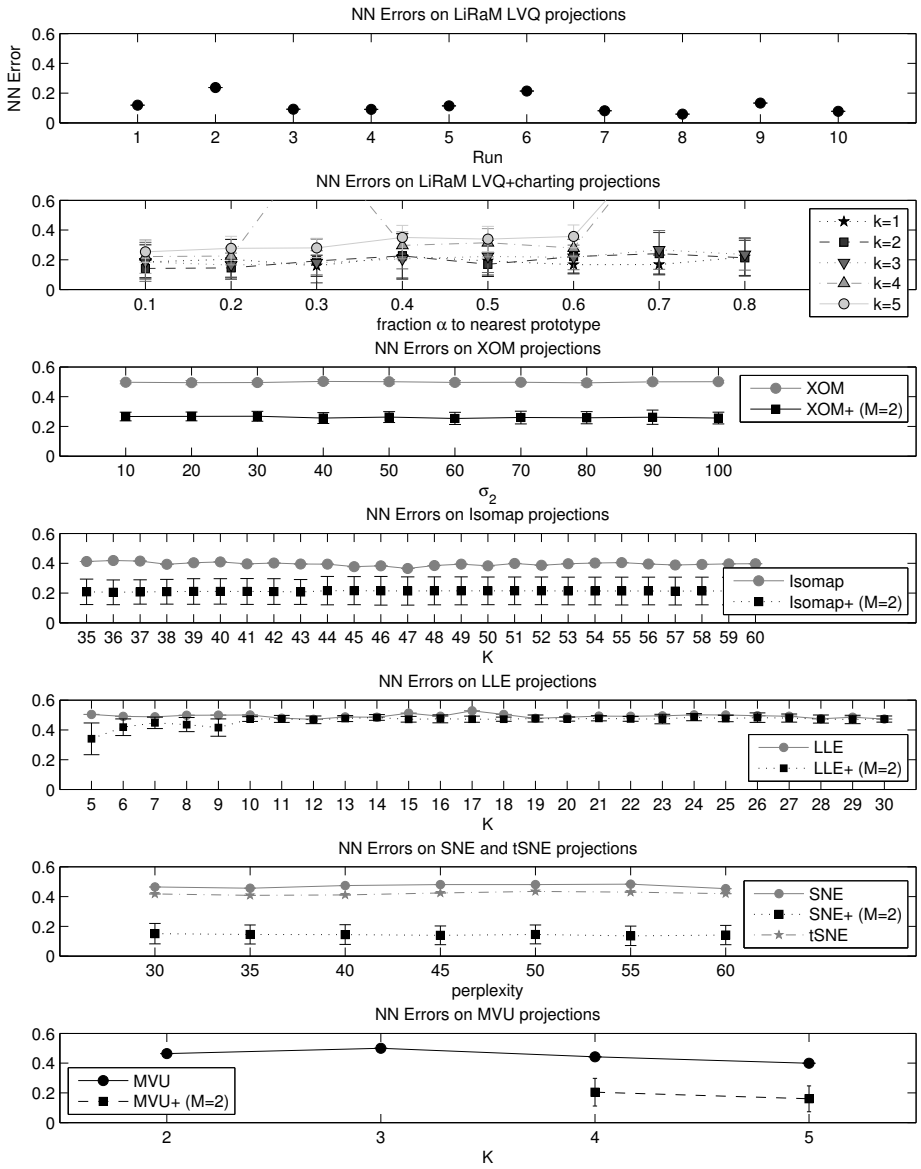


Figure 7.3: 1-NN Errors of the Three Tip Star data set for different methods and parameters. A "+" appended to the name of the method indicates incorporation of local LiRaM LVQ distances with rank M matrices.

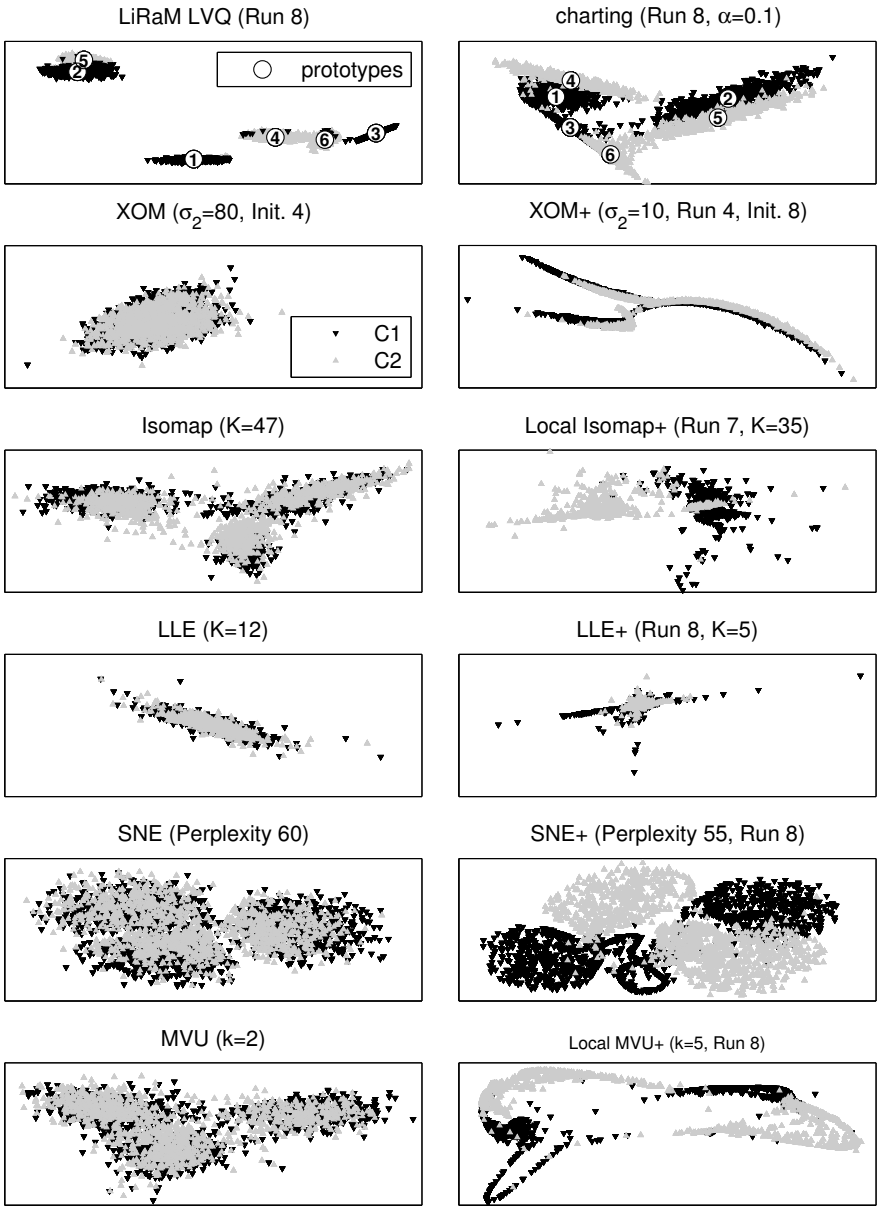


Figure 7.4: Embeddings of the Three Tip Star data set. A "+" appended to the name of the method indicates the incorporation of local LiRaM LVQ.

The 1-NN errors and standard deviations of the two-dimensional projections of all methods with either Euclidean or supervised adapted distance are shown in Fig. 7.3. A "+" appended to the name of a method indicates the use of the learned distance, in addition the reduced target dimension in matrix learning M is given. From top to bottom in Fig. 7.3 the following methods are compared:

1. The 1-NN errors of the LiRaM LVQ projections based on Eq. (7.2) are shown on top. In particular, run 2 and 6 illustrate the problem that regions which are well separated in the original space can be projected onto overlapping areas in low dimension when local projection matrices Ω^j are employed naively. Frequently, however, a discriminative visualization is found, as an example the outcome of run 8 is shown in Fig. 7.4 (upper left panel). Note that the aim of the LiRaM LVQ algorithm is not to preserve topology or distances, but to find projections which separate the classes efficiently. Consequently, clusters four and six, for instance, may be merged in the projection, as they carry the same class label. Nevertheless, the relative orientation of all six clusters persists in the low-dimensional representation.

2. The 1-NN errors of the LiRaM LVQ projections followed by charting with different choices of the responsibilities, cf. σ_j Eq. (7.5). The x-axis corresponds to the factor α which determines σ_j Eq. (7.5) from the mean distance of the k nearest prototypes. Graphs are shown for several values of k, and bars mark the standard deviations observed over the 10 runs. For large α and k the overlap of the local charts increases, yielding larger 1-NN error in the final embedding. Small values of α, k lead to better projection results. The best result is shown in Fig. 7.4 (upper right panel) using $\alpha = 0.1$ and $k = 3$ nearest prototypes. The quality of the projection is not affected by rotations or reflections, consequently the actual positions and orientations of clusters can vary.

3. XOM was trained for $t_{\max} = 50000$ iterations using a learning rate schedule Eq. (6.24) for the image vectors $\boldsymbol{\xi}$ with $\tau_1^{\text{start}} = 0.9$ and $\tau_1^{\text{end}} = 0.05$. The cooperativity function is chosen as Gaussian and like the learning rate $\tau_1(t)$ the variance σ is changed by an appropriate annealing scheme

$$\sigma(t) = \sigma_1 \cdot \left(-\exp\left(\log\left(\frac{\sigma_1}{\sigma_2} \right) / t_{\max} \right) \cdot t \right) \quad . \tag{7.8}$$

   The sampling vectors are initialized randomly in 5 independent runs. The parameter σ_1 is approximately set to the maximum distance in the data space: 1500 and σ_2 is chosen as values between the interval $[10, 100]$. The actual value of σ_2 appears to influence the result only mildly. The 1-NN errors of

the XOM projections with different values of the parameter σ_2 are shown in Fig. 7.3. The incorporation of the trained local distances improves the performance significantly. Example projections are shown in Fig. 7.4 (second row) using Euclidean distances (left panel) and for adaptive distance measure (right panel). The former, unsupervised version cannot handle this difficult data set satisfactorily, while supervised adaptation of the metric preserves the basic structure of the cluster data set.

4. It follows, the 1-NN errors of the Isomap projection with different numbers k of nearest neighbors taken into account. Also here the incorporation of the learned local distance reduces the 1-NN error on the two-dimensional embedding significantly. The parameter k has to be large enough to ensure that a sufficient number of points is connected in the neighborhood graph. Otherwise several subgraphs emerge which are not connected and lead to many missing points in the final embedding. Appropriate example embeddings are shown in Fig. 7.4 in the third row, corresponding to Euclidean distance in the left panel and adaptive metrics in the right panel. In the former, purely unsupervised case, the 3 main clusters are reproduced, but the classes are mixed. When the adaptive distance measure is used, the cluster structure is essentially lost, but the two classes remain separated.

5. The 1-NN errors of the LLE embedding are shown for various numbers k of nearest neighbors considered. LLE displays very limited performance in this data set, hardly any structure is preserved. Even the incorporation of the learned distance measure does not lead to significant improvement, in general. Only for very small values of k the 1-NN error decreases in comparison with the usage of the Euclidean distance. LLE tends to collapse large portions of data onto a single point when the target dimension is too low. Hence, even a small 1-NN error may not indicate a good and interpretable visualization. The best embeddings are shown in Fig. 7.4 in the forth row.

6. The 1-NN errors of SNE and t-SNE are slightly better than the other unsupervised methods. Both methods preserve the main cluster structure, but not the class memberships. Like already observed with Isomap$^+$ also with the supervised version of SNE (SNE$^+$) the cluster structure is essentially lost, but the two classes are separated as much as possible and a remarkable increase in the 1-NN error of the embedded data is observed. Example embeddings are shown in Fig. 7.4 (fifth row) and for t-SNE in Fig. 7.2 right panel.

7. The 1-NN errors of MVU are comparable to the SNE and t-SNE results. Like them the main cluster structure is visible, but not the class memberships. In the

supervised variant MVU$^+$ the cluster structure is essentially lost as observed with Isomap$^+$ and SNE$^+$ too, but the two classes are separated relatively well. This leads to a remarkable decrease in the 1-NN error of the embedded data points. The best embeddings are shown in Fig. 7.4 bottom row.

Note that, due to the presence of only two classes, standard LDA would yield a projection to one dimension only. We have also applied kernel PCA with Gaussian kernel and different values of σ, but we obtained only poor 1-NN errors on the embedded data with a best value of about 41%. As expected, purely unsupervised methods preserve hardly any class structure in the obtained projections. For several methods, however, the performance with respect to discriminative low-dimensional representation can be improved dramatically by taking into account label information in the local distance measures.

Figure 7.5 shows the computation times vs. the number of points to be embedded of different dimension reduction techniques on the Three Tip Star data set. We only measure the time necessary to embed the data after learning the local metrics

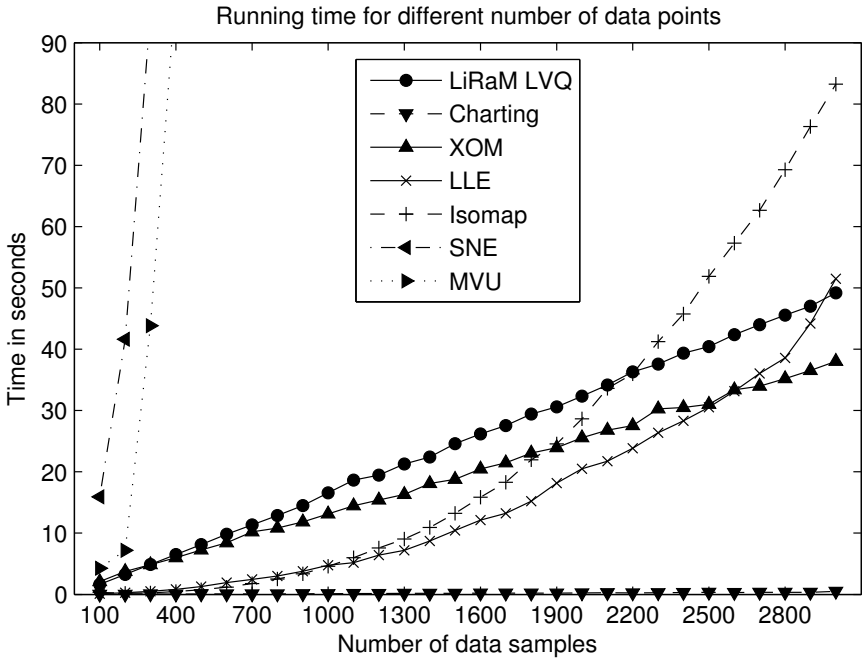


Figure 7.5: The running time of different dimension reduction methods depending on the number of samples to embed.

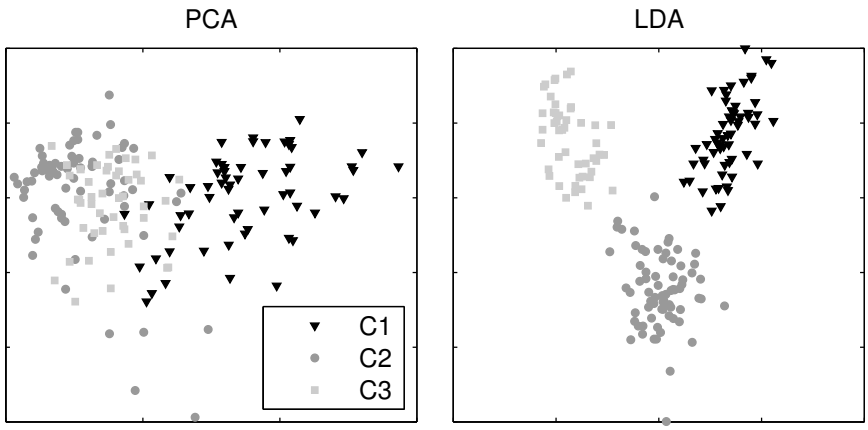


Figure 7.6: Example embeddings of the Wine data set for PCA and LDA.

with LiRaM LVQ. The algorithms were performed on the same Windows XP 32bit version machine[4] using Matlab R2008b. The LiRaM LVQ algorithm was applied using six prototypes and 100 epochs. The other parameters were chosen as mentioned above. The charting technique uses the six local linear projections provided by the LVQ approach with responsibilities computed by Eq. (7.4). XOM is trained for 1500 steps and above mentioned parameters, LLE uses $k = 35$, Isomap $k = 35$ and MVU $k = 3$ nearest neighbors. SNE was performed with a perplexity of 30. The LVQ based approach, charting and XOM show a linear relationship between the number of points and the necessary computation time, whereas the other methods show quadratic or even worse complexity.

7.3.2 Wine data set

The wine data from (Aeberhard et al. 1992) available at (Asuncion et al. 1998) contains 178 samples in 13 dimensions divided in three classes. As proposed in (Rogers and Girolami 2007) we first transformed the data to have zero mean and unit variance features. Maximum Likelihood Estimation (MLE) (Levina and Bickel 2005) approximate the intrinsic dimension to 4. We set the reduced target dimension to two. PCA achieves a leave-one-out 1-NN error of 28% in the mapped data set. In comparison, supervised LDA (Fukunaga 1990) leads to a relatively small 1-NN error of 1%. Fig. 7.6 shows the two-dimensional representations of the data set obtained by PCA and LDA, respectively.

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

Localized LiRaM LVQ was trained for $t_{\max} = 300$ epochs, with one prototype per class. Each prototype was initialized close to class centers, elements of the matrices Ω^j were initialized with values between $[-1, 1]$ with subsequent normalization. The learning rate for prototype updates follows the schedule $\tau_1(t) = 0.1/(1+(t-1)\cdot 0.01)$; metric learning starts at epoch $t_M = 30$ with the learning rate $\tau_2(t) = 0.01/(1 + (t - 50) \cdot 0.001)$. We run the localized LiRaM LVQ 10 times with random initializations and with rank $M = 2$ and $M = 4$ of the relevance matrices, respectively. In all runs we observe 100% correct classification for this data set. The resulting matrices are used to embed the data into the two-dimensional space. In order to compare the different approaches we compute the 1-NN errors in the projected data under various parameter settings, results are shown in Fig. 7.7 and the best projections can be found in Fig. 7.8. The incorporation of trained distances in some unsupervised methods are indicated by a "+" appended to the name, together with the maximum rank M.

1. In the direct LVQ-based mapping following Eq. (7.2), two prototypes project into the same area in some of the runs, but most runs result in a clear separation of the three classes. The charting technique is combined with the three local projections obtained from the localized LiRaM LVQ ($M = 2$) and computed with various parameters α to fix the responsibilities (see Eq. (7.4)). A reasonable overlap of the local projections is required: If α is chosen too small the 1-NN error displays large variations in the runs. For this data set a value of $\alpha = 0.4$ is sufficiently large to yield discriminative visualizations.

2. XOM was trained like with the previous data set for $t_{\max} = 50000$ iterations with the same learning rate schedule for τ_1 Eq. (6.24) and σ Eq. (7.8). The parameter σ_1 is set to 2 and σ_2 to 0.15. The sampling vectors are initialized randomly in 10 independent runs. The results of XOM and XOM$^+$ in combination with adaptive local distances are analogous to those for the Three Tip Star data. The improvement due to the incorporation of label information through the distance measure is even more significant, the method yields very small 1-NN errors in the Wine data set.

3. The k-Isomap with Euclidean distance performs worse on this data set with an 1-NN error of about 30%. With the incorporation of the learned distance measure and a sufficiently large neighborhood value k all mappings separate the classes very well. For smaller values of k the neighborhood graph is not connected. In the worst case the procedure yields three unconnected subgraphs, where only samples are connected which belong to the same prototype. When all samples are connected the approach is very robust and shows no variation with respect to the LVQ run.

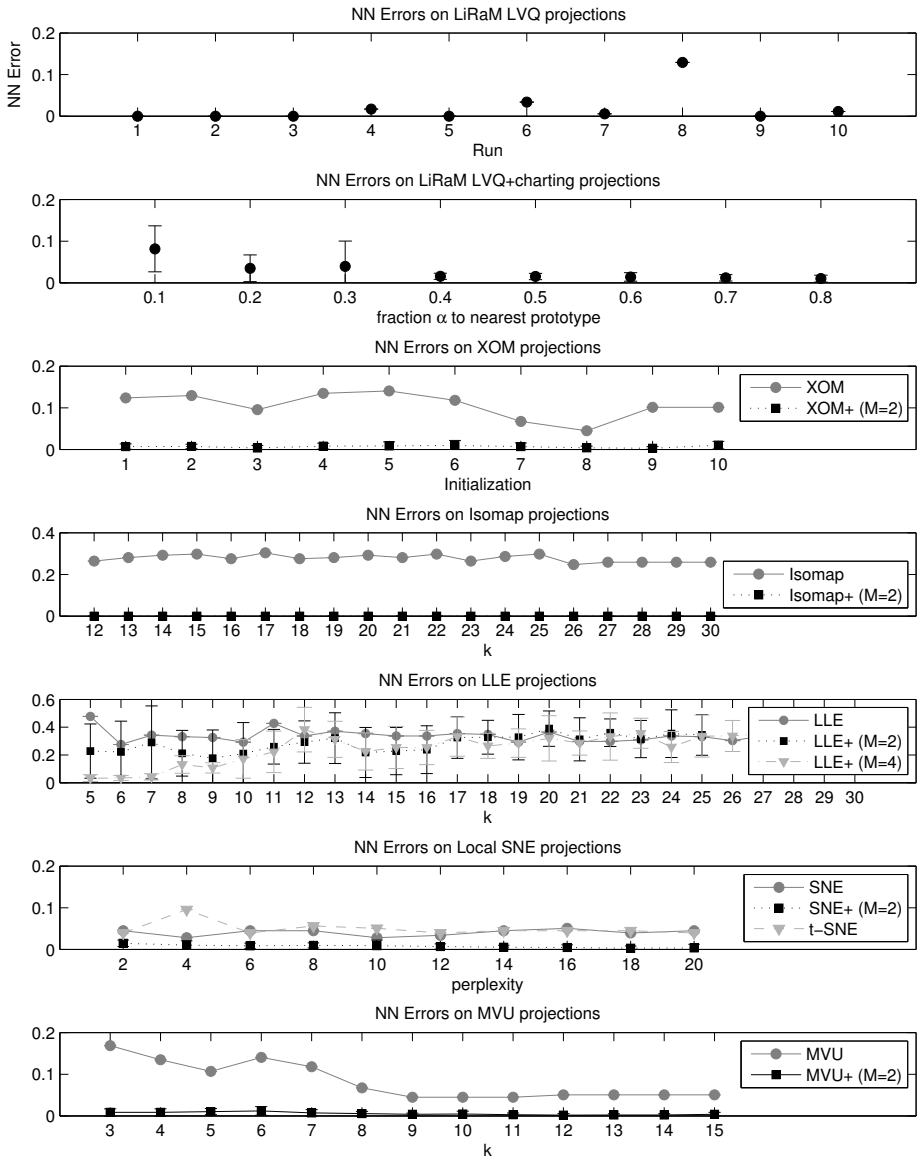


Figure 7.7: 1-NN Errors of the Wine data set for different methods and parameters. A "+" appended to the name of the method indicates incorporation of local LiRaM LVQ distances with rank M matrices.

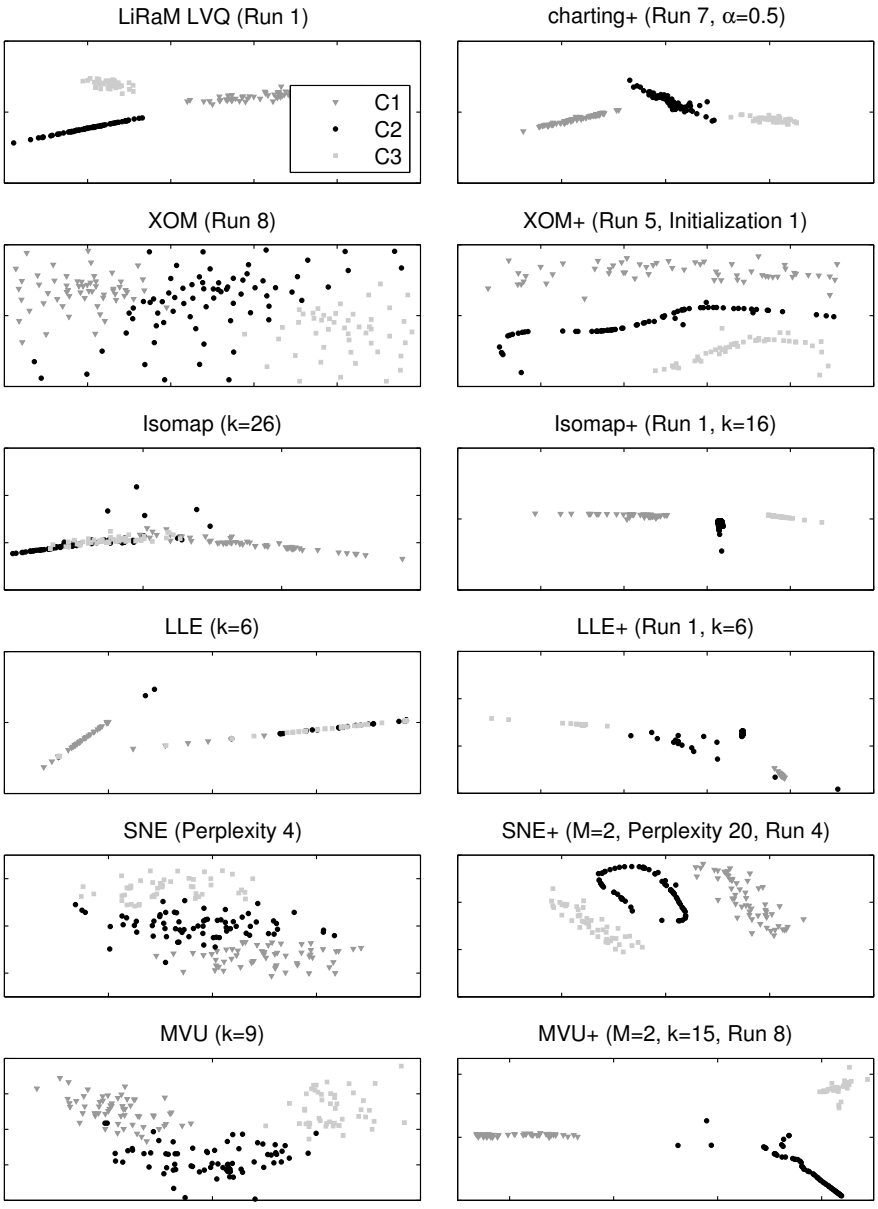


Figure 7.8: Example embeddings of the Wine data set. A "+" appended to the name of the method indicates the incorporation of local LiRaM LVQ distances.

4. The performance of LLE depends strongly on the number k of nearest neighbors taken into account. For large k the advantage of using a supervised learned distance measure essentially vanishes. The variations between different runs are particularly pronounce for rank $M = 2$ and no significance improvement over the purely unsupervised LLE is achieved. However, for small k (e.g. $k = 5, 6, 7$) and with rank $M = 4$ very low 1-NN errors are obtained.

5. The SNE and t-SNE show already in the unsupervised versions good results as shown in Fig. 7.7. The 1-NN error is not that much dependent on the chosen perplexity, only slight changes can be observed. With the incorporation of the learned distance measure the visualizations can be improved further and the dependence on the perplexity is even less.

6. The unsupervised MVU showed a strong dependence on the number k of neighbors taken into account. With a sufficient big k the algorithm show already good results when it is used in an unsupervised way. The incorporation of the class labels however shows only a weak dependence on the number of neighbors and in most of the results the classes are perfectly separated.

7.3.3 Segmentation

The Segmentation data set (available at the UCI repository (Asuncion et al. 1998)) consists of 19 features which have been constructed from randomly drawn regions of 3×3 pixels in a set of 7 manually segmented outdoor images. Every sample is assigned to one of seven classes: brickface, sky, foliage, cement, window, path and grass (referred to as C1, ..., C7). The set consists of 210 training points with 30 instances per class and the test set comprises 300 instances per class, resulting in 2310 samples in total. We did not use the features (3,4,5) as they display zero variance over the data set. For preprocessing we normalized the data by a z-Transformation resulting in zero mean and unit variance features. An Maximum Likelihood estimation yields an intrinsic dimension of about 3, so we use rank limits of $M \in \{2, 3\}$ for the computation of the local distances in this data set. LDA yields a classification error of approx. 20% for a projection into two dimensions while PCA displays a 1-NN error of 31%.

Localized LiRaM LVQ was trained for $t_{\max} = 500$ epochs, with one prototype per class. Each prototype was initialized close to class center, and elements of the matrices Ω^j are drawn randomly from $[-1, 1]$ according to a uniform density with subsequent normalization of the matrices. The learning rate for prototypes follows the schedule $\tau_1(t) = 0.01/(1 + (t - 1) \cdot 0.001)$. Metric adaptation starts at epoch $t_M = 50$ with learning rates $\tau_2(t) = 0.001/(1 + (t - 50) \cdot 0.0001)$. We run localized

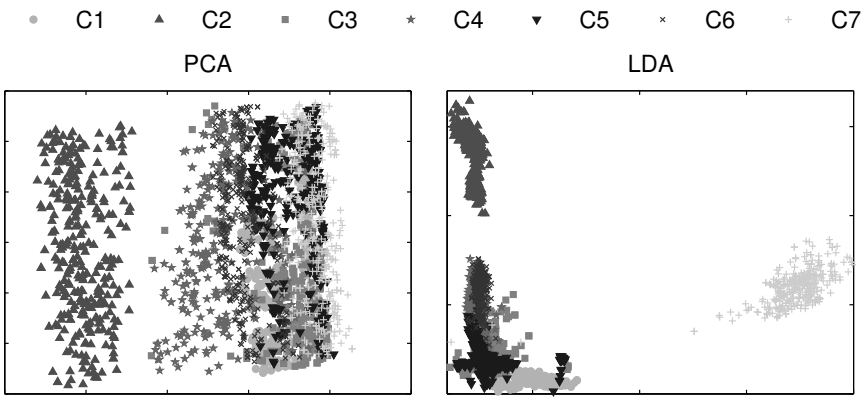


Figure 7.9: Example embeddings of the Segmentation data set for PCA and LDA.

LiRaM LVQ 10 times with random initialization and with a rank limit of $M = 2$ and $M = 3$, respectively. For $M = 2$ we achieve a mean classification error of about 8% in all runs and with $M = 3$ the mean classification error is 7%. The obtained 1-NN errors are shown in Fig. 7.10 and some example visualizations are given in Fig. 7.11.

1. The quality of direct LiRaM LVQ projections vary from run to run. One favorable projection is shown in Fig. 7.11 in the first row on the left side. The classes C2 and C6 are well separated with large distances from the other classes. Also, most samples of C4 and C1 are clustered properly, while class C3 is spread and overlaps with class C7. This outcome is not too surprising, since C3 and C7 correspond to foliage and grass, respectively, two classes that may be expected to have similar characteristics in feature space.

2. In the combination with a charting step results are rather robust with respect to the parameter settings (α, k). Here, the best result is achieved with $\alpha = 0.1$ and $k = 1$ (Fig. 7.11, top right panel). Again, three classes are well separated from the others. The remaining four classes are projected into a relatively small area. Three of these classes are very close: window, brickface, and cement.

3. XOM was trained for $t_{\max} = 50000$ iterations with the same learning rate schedule for τ_1 and σ like for the other data sets. We set the parameters to $\tau_1^{\text{start}} = 0.9$, $\tau_1^{\text{end}} = 0.05$ and σ_1 to nearly the maximum distance in the data space: 1500 and σ_2 is chosen as values between the interval $[5, 70]$. The sampling vectors are initialized randomly in 5 independent runs. In the application of XOM we observe once more a clear improvement when incorporating

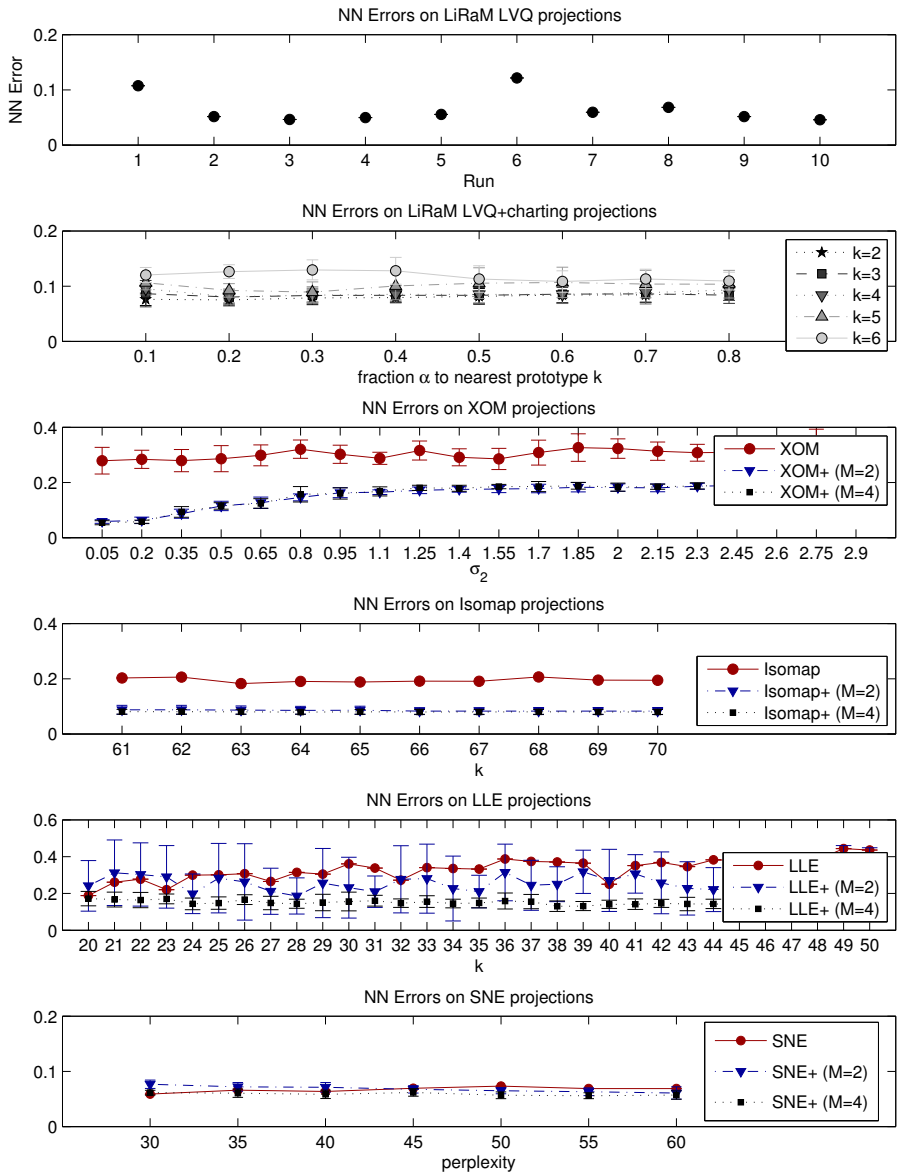


Figure 7.10: 1-NN Errors of the Segmentation data set for different methods and parameters. A "+" appended to the name of the method indicates incorporation of local LiRaM LVQ distances with rank M matrices.

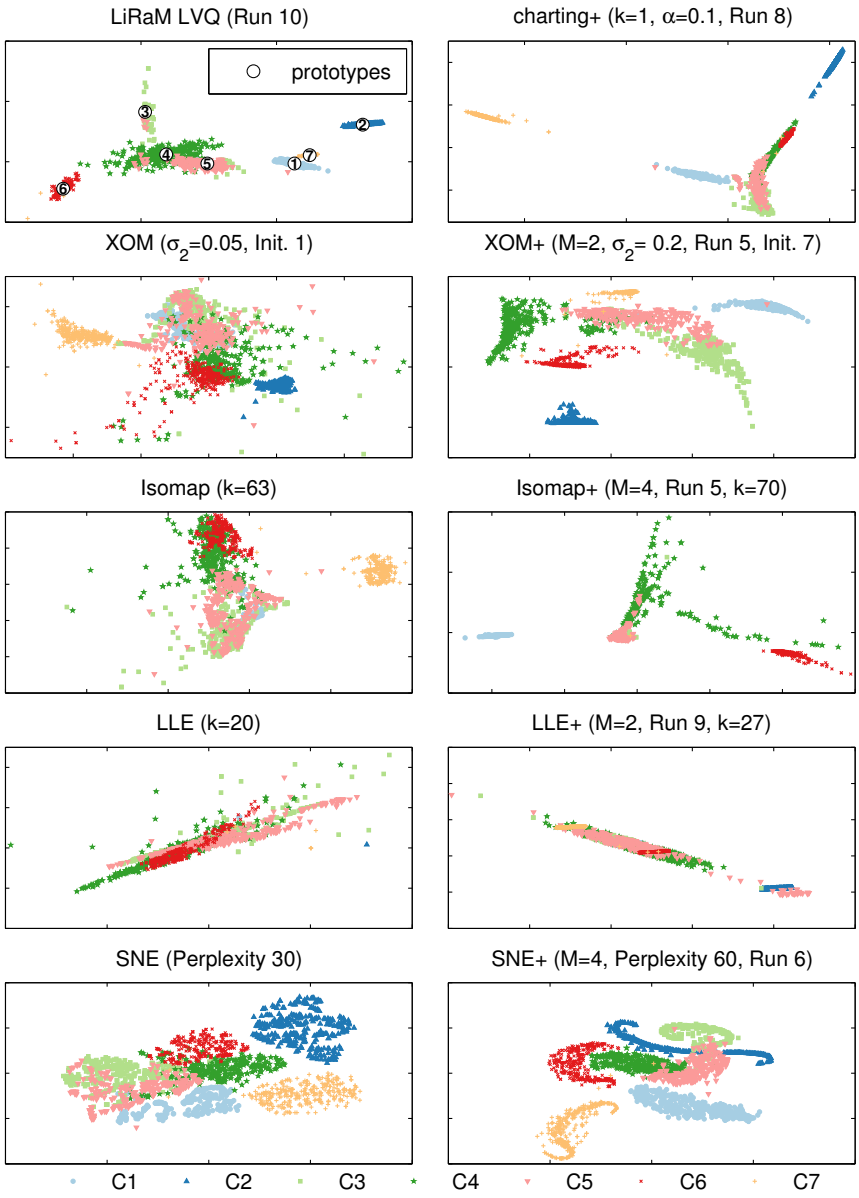


Figure 7.11: Example embeddings of the Segmentation data set. A "+" appended to the name of the method indicates the incorporation of local LiRaM LVQ distances.

the adaptive local metrics obtained in LiRaM LVQ. Example projections are shown in Fig. 7.11 (second row).

4. For Isomap a minimum value of $k \geqslant 48$ is necessary to obtain fully connected neighborhood graphs and, hence, embed all points. The incorporation of adaptive local distances leads to a clear improvement of the 1-NN error in the mapping. As expected, a low rank M of the local matrices results in inferior 1-NN errors if M is smaller than the intrinsic dimension of the data. When incorporating adaptive distances with very large k, a fully connected graph can be obtained and all data are mapped. However, then, closer classes would highly overlap in the projections and the visualization would not be discriminative. If, on the other hand, a smaller k is chosen, some of the classes are absent in the graph and, consequently, in the visualization. As a consequence of this effect, in Fig. 7.11 (third row, right panel) class C2 subgraph is absent.

5. Like in the previous examples, LLE performs relatively poor. The 1-NN error can be decreased by using adaptive distances but points tend to be collapsed in the projection due to the discriminative nature of the distance measure. Most visualizations with relatively low 1-NN errors display an almost linear arrangement of all classes, cf. Fig. 7.11 (fourth row, left panel). An example visualization after incorporation of adaptive metrics is shown in the right panel. While the visualization appears to be be better, qualitatively, the above mentioned basic problem of LLE persists.

6. The last row of Fig. 7.11 displays the two-dimensional representations provided by SNE and SNE$^+$ for perplexities in the interval [30 60]. The unsupervised variant performs already quite well, but the incorporation of the learned local distances improves it even further especially for higher perplexities and bigger values for the limited rank M of the LiRaM LVQ algorithm (see Fig. 7.10).

Classes C2 (sky) and C7 (grass) are obviously separable by all applied methods, both unsupervised and supervised. On the other hand, the discrimination of classes C4 (foliage) and C5 (window) appears to be difficult, in particular in unsupervised dimension reduction. Since the patches are randomly drawn from the images they might contain pixels belonging to more than one class and the overlap is comprehensible to some extend.

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

with smaller values of k. The provided code demands a fully connected graph, so the number of constraints of the SDP becomes too large to be solved in reasonable time and needs more memory than we have.

7.3.4 USPS Digits

The USPS (United States Postal Service) data set consists of images of hand written digits of a resolution of 16×16 pixel. We normalized the data to have zero mean and unit variance features and used a test set containing 200 observations per class. Since it is a digit recognition task , we have the classes $\in [0, \ldots, 9]$ resulting in 2000 samples for the embedding. The 1-NN errors of all compared methods are shown in Fig. 7.12.

Localized LiRaM LVQ was trained for $t_{\max} = 500$ epochs, with one prototype per class and the same initialization scheme for the prototypes and matrices, learning rates and learning schedules like explained in Section 7.3.3.

1. The direct LiRaM LVQ projections separate the classes nearly perfectly and one favorable projection is shown in Fig. 7.13 in the first row on the left side.

2. In the combination with a charting step the best result is achieved with $\alpha = 0.1$ and $k = 2$ (Fig. 7.13, top right panel). Four classes appear to be squeezed together, but the overlap is still small if zoomed.

3. XOM was trained in the same way like mentioned in Section 7.3.3 with σ_2 chosen as values between the interval [0.01,2]. The incorporation of the adaptive local metrics obtained in LiRaM LVQ once more improve the results of the XOM dramatically. Example projections are shown in Fig. 7.13 (second row).

4. For Isomap the incorporation of adaptive local distances improves the 1-NN error in the mapping. Like mentioned with the other data sets some data points appear to be too separated from the others if the local distances are used, so the mapping may miss them with a small neighborhood parameter k. Like in the previous examples, LLE performs relatively poor, but can be enhanced by using the local dissimilarities given by LiRaM LVQ (Fig. 7.13, fourth row).

5. SNE performs relatively well, but t-SNE showed a remarkable better 1-NN error on this data set. Still the class structure is hardly recognizable on the unsupervised mapping, while it becomes clear if the local distances are incorporated (Fig. 7.13, fifth row, right panel). The supervised SNE$^+$ results in 10 nicely recognizable clusters.

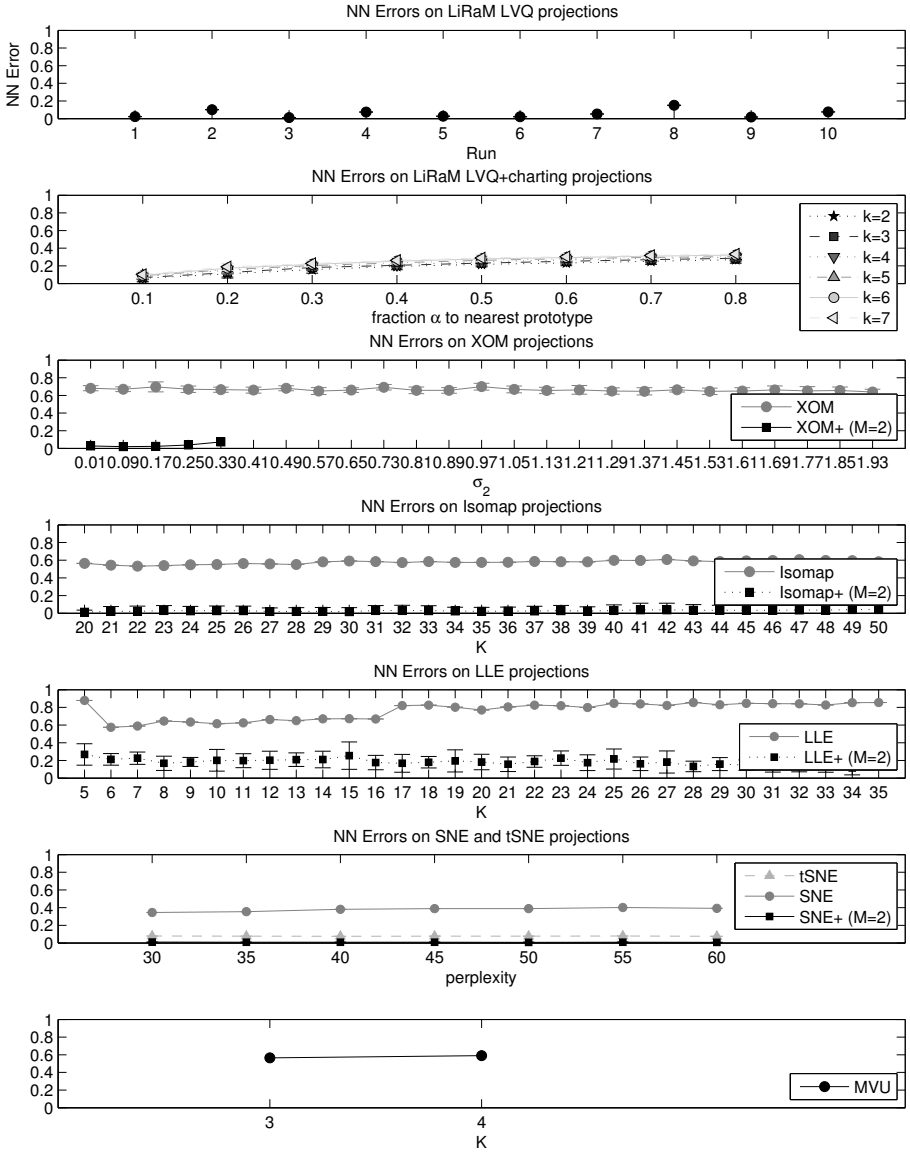


Figure 7.12: 1-NN Errors of the USPS Digits data set for different methods and parameters. A "+" appended to the name of the method indicates incorporation of local LiRaM LVQ distances with rank M matrices.

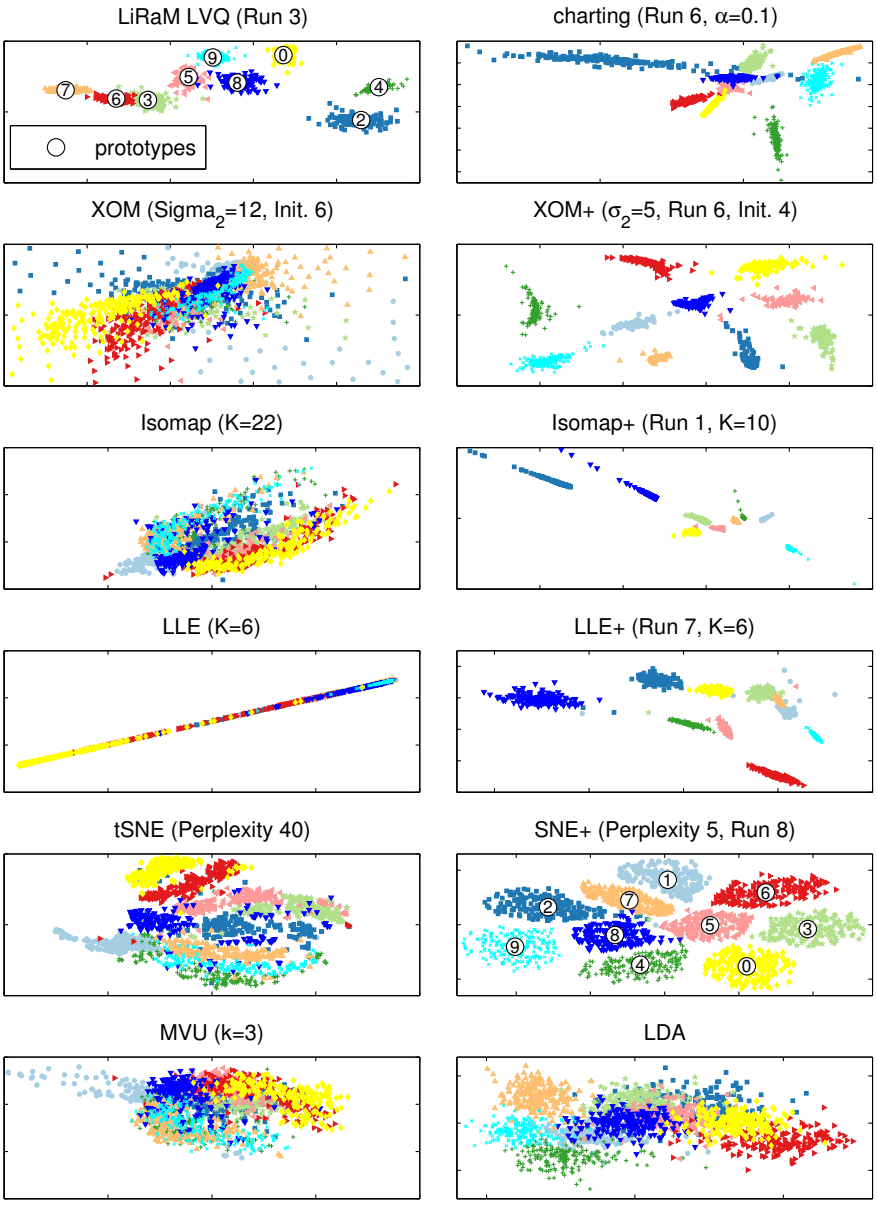


Figure 7.13: Example embeddings of the USPS Digits data set. A "+" appended to the name of the method indicates the incorporation of local LiRaM LVQ.

6. The last row of Fig. 7.13 displays the two-dimensional representations provided by MVU and LDA. MVU does not perform very well on this data set and LDA yields a classification error of 35%. We could not apply MVU with incorporation of the local distances provided by LiRaM LVQ, because the classes are separated so well in this case that a huge value of nearest neighbors k would be necessary to get a connected graph.

7.4 Conclusions

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

Interestingly, the resulting methods have quite different complexity: while charting uses the fact that information is compressed in the prototypes resulting in an only linear scheme depending on the number of data, LLE, SNE, and Isomap end up with quadratic or even cubic complexity. Further, charting techniques and similar provide the only methods in this collection which yield an explicit embedding map rather than an embedding of the given points only. The behavior of the resulting discriminative visualization techniques has been investigated in one artificial and three real life data sets. The best results for all methods and data sets are summarized in Table 7.1. According to the different objectives optimized by the visualization techniques, the results are quite diverse and no single method which is optimum for every case can be identified. In general, discriminative visualization as introduced in this paper improves all the corresponding unsupervised methods and also alternative state-of-the-art schemes such as t-SNE. Further, the techniques presented in this Chapter are superior to discriminative LDA which is restricted to linear embedding. It seems that charting offers a good choice in many cases, in particular since it is a method with only linear effort which provides an explicit embedding

Table 7.1: 1-NN errors (and Standard deviation) on the different data sets.

Method	3 Tip Star	Wine	Segmentation	USPS Digits
LiRaM LVQ	0.06 (0.0)	0.00 (0.0)	0.07 (0.0)	0.01 (0.0)
charting	0.14 (0.1)	0.01 (0.0)	0.13 (0.0)	0.06 (0.0)
XOM	0.49 (0.0)	0.04 (0.0)	0.25 (0.0)	0.64 (0.0)
XOM+(M=2)	0.25 (0.0)	0.00 (0.0)	0.11 (0.0)	0.02 (0.0)
XOM+(M=3)	-	-	0.11 (0.0)	-
Isomap	0.36 (0.0)	0.25 (0.0)	0.23 (0.0)	0.53 (0.0)
Isomap+(M=2)	0.20 (0.1)	0.00 (0.0)	0.18 (0.1)	0.01 (0.0)
Isomap+(M=3)	-	-	0.13 (0.1)	-
LLE	0.47 (0.0)	0.28 (0.0)	0.36 (0.0)	0.57 (0.0)
LLE+(M=2)	0.34 (0.1)	0.18 (0.2)	0.25 (0.1)	0.11 (0.1)
LLE+(M=3)	-	0.03 (0.0)	0.19 (0.0)	-
SNE	0.45 (0.0)	0.03 (0.0)	0.11 (0.0)	0.34 (0.0)
SNE+(M=2)	0.14 (0.1)	0.00 (0.0)	0.10 (0.0)	0.01 (0.0)
SNE+(M=3)	-	-	0.09 (0.0)	-
t-SNE	0.41 (0.0)	0.04 (0.0)	0.85 (0.0)	0.08 (0.0)
MVU	0.40 (0.0)	0.04 (0.0)	-	0.56 (0.0)
MVU+(M=2)	0.16 (0.1)	0.00 (0.0)	-	-
LDA	-	0.01 (0.0)	0.20 (0.0)	0.35 (0.0)

map.

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

Published as:

K. Bunte, B. Hammer, T. Villmann, M. Biehl and A. Wismüller – *"Neighbor Embedding XOM for Dimension Reduction and Visualization,"* Neurocomputing, vol. 74, no. 9, pp. 1340–1350, 2010.

K. Bunte, B. Hammer, T. Villmann, M. Biehl and A. Wismüller – *"Exploratory Observation Machine (XOM) with Kullback-Leibler Divergence for Dimensionality Reduction and Visualization,"* in Proc. of European Symposium on Artificial Neural Networks (ESANN), pp. 87–92, Bruges, Belgium, April 2010.

Chapter 8

Self Organized Neighbor Embedding for Dimension Reduction and Visualization

The important thing in science is not so much to obtain new facts as to discover new ways of thinking about them.

Sir William Bragg (1862 - 1942)

Abstract

We present an extension of the Exploration Observation Machine for structure-preserving dimensionality reduction. Based on minimizing the Kullback-Leibler divergence of neighborhood functions in data and image spaces, this Self Organized Neighbor Embedding (SONE) creates a link between fast sequential online learning known from topology-preserving mappings and principled direct divergence optimization approaches. We quantitatively evaluate our method on real world data using multiple embedding quality measures. In this comparison, SONE performs as a competitive trade-off between high embedding quality and low computational expense, which motivates its further use in real-world settings throughout science and engineering.

8.1 Introduction

Various dimension reduction techniques have been introduced based on different properties of the original data to be preserved. A detailed description of an handpicked amount of unsupervised and supervised methods can be found in Chapter 6. For a comprehensive review on nonlinear dimensionality reduction methods, we refer to (Lee and Verleysen 2007). In Chapter 3 and 7 we proposed further methods for supervised linear and non-linear dimension reduction and visualization.

Recently, a novel computational approach to topology learning has attracted attention for advanced data processing: The Exploration Observation Machine (XOM) (Wismüller 2006, Wismüller 2009d, Wismüller 2009b, Wismüller 2001, Wismüller 2011) (and references therein) systematically reverses the data-processing work flow in topology-preserving mappings. By consistently exchanging functional and structural components of topology-preserving mappings, XOM can be seen as a computational framework that computes graphical representations of high-dimensional observations by a strategy of self-organized model adaptation. Although simple and computationally efficient, XOM enjoys a surprising flexibility to simultaneously contribute to several different domains of advanced machine learning, scientific data analysis, and visualization. In particular, it supports both structure-preserving dimensionality reduction and data clustering.

The complexity of most non-linear dimension reduction techniques grows at least quadratically with the number of points to embed. The aim of Self Organized Neighbor Embedding (SONE) proposed in this Chapter is to create a conceptual link between fast sequential online learning known from topology-preserving mappings and principled direct divergence optimization approaches, such as Stochastic Neighbor Embedding (SNE) and t-distributed SNE (t-SNE). So it can be seen as a trade-off between low computational costs and high quality of the final embedding. The complexity is linear with the number of points and can be easily controlled by the user. Furthermore, prior knowledge and task specific requirements can be incorporated to the embedding result.

We will describe the basic XOM algorithm and the SONE extension in Section 8.2 and Section 8.3. We discuss the parameters in section 8.4 and furthermore we spend some words on the complexity in comparison with other techniques in Section 8.5, discuss the embedding results on two benchmark data sets in Section 8.6, and conclude in Section 8.7.

8.2 The Exploratory Observation Machine

XOM maps a finite number of high-dimensional data points $\boldsymbol{x}^i \in \mathcal{X}$ in the observation space $\mathcal{X}$ to low-dimensional image vectors $\boldsymbol{\xi}^i \in \mathcal{E}$ in the embedding space $\mathcal{E}$. The embedding space is associated with a structure hypothesis, given by a number of sampling vectors $\boldsymbol{s} \in \mathcal{E}$, which corresponds to the final structure in which the data is embedded. These can be seen as a generalization of the prototypes as included in the Self-organizing Map (SOM). Reasonable choices for the sampling vectors $\boldsymbol{s}$ are: the location on a regular lattice structure in $\mathcal{E}$, discrete positions in $\mathcal{E}$ as representation of a finite number of class centers, drawn from a mixture of Gaussian

to represent a finite number of clusters, or uniformly sampled in a region of $\mathcal{E}$ to indicate that the visualization of the data should occupy the full projection space. Unlike SOM, XOM does not project the sampling vectors $\boldsymbol{s}$ to the data space, rather it projects the data to the embedding space. Nevertheless, the sampling vectors define receptive fields by a decomposition into points mapped closest to the sampling vectors. An approximate back projection of the sampling vector can be defined as the best match input vector

$$\Psi(\boldsymbol{s}) = \boldsymbol{x}^i \text{ where } d_{\mathcal{E}}(\boldsymbol{s}, \boldsymbol{\xi}^i) \text{ is minimum.} \tag{8.1}$$

The images $\boldsymbol{\xi}^i$ are initialized randomly and adapted iteratively during the training triggered by the structure of the embedding space. All $\boldsymbol{\xi}^i$ are adapted into the direction of the actual $\boldsymbol{s}$ according to the distances between the best match input $\Psi(\boldsymbol{s})$ and their counterparts $\boldsymbol{x}^i$ in the observation space $\mathcal{X}$. For a given sampling vector $\boldsymbol{s}$ the adaptation rule is given by:

$$\boldsymbol{\xi}^k := \boldsymbol{\xi}^k - \tau \cdot h_\sigma(d_{\mathcal{X}}(\Psi(\boldsymbol{s}), \boldsymbol{x}^k)) \frac{\partial d_{\mathcal{E}}(\boldsymbol{s}, \boldsymbol{\xi}^k)}{\partial \boldsymbol{\xi}^k} \quad , \tag{8.2}$$

where $\tau > 0$ denotes the learning rate, $d_{\mathcal{X}}$ refers to the distance in the observation space, e.g. the Euclidean distance and

$$h_\sigma(d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{x}^j)) = h_\sigma^{ij} = \exp\left(\frac{-d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{x}^j)}{2\sigma^2}\right) \text{ with } \sigma > 0 \tag{8.3}$$

defines the neighborhood cooperation. In this way the projections $\boldsymbol{\xi}$ are arranged around the priorly chosen structure elements $\boldsymbol{s}$ such that image vectors are close to the same sampling vector if their corresponding data points $\boldsymbol{x}$ are neighbored in the data space. The method is summarized in Algorithm 8.1:

Algorithm 8.1 : Exploratory Observation Machine (XOM)

1: choose a structure hypothesis, given by sampling vectors $\boldsymbol{s} \in \mathcal{E}$
2: initialize the image vectors $\boldsymbol{\xi}$, e.g. randomly or by means of a PCA.
3: compute the neighborhood function , e.g. a Gaussian Eq. (8.3)
4: **while** stopping criterion not reached **do**
5: present a sampling vector $\boldsymbol{s}$ from the structure hypothesis
6: find the best matching input vector following Eq. (8.1)
7: perform the update of all image vectors with the adaptation rule Eq. (8.2)
8: **end while**

8.2.1 Formalization of a cost function

As the SOM, XOM in its original form does not correspond to a cost function. However, as proposed in (Bunte, Hammer, Villmann, Biehl and Wismüller 2010), a variation following (Heskes 1999) by setting the best match input data vector to the average

$$\Psi(\boldsymbol{s}) = \boldsymbol{x}^i \text{ where } \sum_j h_\sigma(d_\mathcal{X}(\boldsymbol{x}^i, \boldsymbol{x}^j)) d_\mathcal{E}(\boldsymbol{s}, \boldsymbol{\xi}^j) \text{ is minimum } . \tag{8.4}$$

This leads to the cost function:

$$E_{\mathrm{XOM}} \sim \int \sum_i \delta_{\Psi(\boldsymbol{s}),\boldsymbol{x}^i} \cdot \sum_{j=1}^{N} h_\sigma(d_\mathcal{X}(\boldsymbol{x}^i, \boldsymbol{x}^j)) \cdot d_\mathcal{E}(\boldsymbol{s}, \boldsymbol{\xi}^j)\, p(\boldsymbol{s})\, d\boldsymbol{s}, \tag{8.5}$$

where δ denotes the Kronecker delta. The derivative of E_{XOM} with respect to $\boldsymbol{\xi}^k$ can be found in 8.A and yields the XOM learning rule given in Eq. (8.2). Thus, XOM tries to minimize the distortion of sampling vectors $\boldsymbol{s}$ and projections $\boldsymbol{\xi}^j$ whereby this term is weighted according to a Gaussian function depending on the distance of the inverse images $\Psi(\boldsymbol{s})$ and $\boldsymbol{x}^j$ in the data space.

8.3 SONE using generalized Kullback-Leibler

XOM, unlike SNE and many other embedding algorithms, exhibits the interesting property that it allows to impose a prior structure on the projection space, which is a property that can also be found in SOM. Like many other visualization techniques, SNE has a computational and memory complexity that grows quadratically with the number of data points, because it bases on the computation of pairwise affinities in the projection space (for detailed description see Algorithm 6.4 in Section 6.2). The complexity of XOM can be easily controlled by the structure definition and is linear with the number of data points and the number of sampling vectors. We propose to combine the ideas of XOM with the concept of direct divergence optimization as proposed by SNE, to merge the advantages of both methods.

By means of the cost function Eq. (8.5) we are able to define new learning rules for the XOM algorithm based on the generalized Kullback-Leibler (GKL) divergence for not normalized positive measures p and q with $0 \leqslant p, q \leqslant 1$:

$$\mathrm{D}_{\mathrm{GKL}}(p\|q) = \int \left[p(\boldsymbol{x}) \log\left(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right)\right] d\boldsymbol{x} - \int \left[p(\boldsymbol{x}) - q(\boldsymbol{x})\right] d\boldsymbol{x} \ . \tag{8.6}$$

We consider the use of normalized and symmetrized probability densities (proposed for SNE) as unnecessary restriction and define our concept in a more general way.

In contrast to (Villmann and Haase 2011, Mwebaze et al. 2011), however, we do not use the GKL divergence as a distance measure *within* the original or the embedding space, but as a dissimilarity measure *between* the two spaces. The cooperativity functions $h_\sigma(d_\mathcal{X}(\boldsymbol{x}^i, \boldsymbol{x}^j))$ and $g_\varsigma(d_\mathcal{E}(\boldsymbol{s}, \boldsymbol{\xi}^j))$ used as positive measures, can be defined analogously to Eq. (8.3):

$$h_\sigma^{\boldsymbol{x}^i}(k) = h_\sigma(d_\mathcal{X}(\boldsymbol{x}^i, \boldsymbol{x}^k)) = \exp\left(\frac{-d_\mathcal{X}(\boldsymbol{x}^i, \boldsymbol{x}^k)}{2\sigma^2}\right) \tag{8.7}$$

$$g_\varsigma^{\boldsymbol{s}}(k) = g_\varsigma(d_\mathcal{E}(\boldsymbol{s}, \boldsymbol{\xi}^k)) = \exp\left(\frac{-d_\mathcal{E}(\boldsymbol{s}, \boldsymbol{\xi}^k)}{2\varsigma^2}\right) \quad . \tag{8.8}$$

They model the neighborhoods in the original space and the embedding space, similar to the probability densities p and q in the SNE formulation. Following the ideas of t-SNE (see Algorithm 6.5 in Section 6.2) the neighborhood function of the embedding space $g_\varsigma(d_\mathcal{E}(\boldsymbol{s}, \boldsymbol{\xi}^j))$ could be chosen as a heavy-tailed distribution, e.g. the Student-t-distribution similar to Eq. (6.10):

$$g_\varsigma^{\boldsymbol{s}}(k) = (1 + d_\mathcal{E}(\boldsymbol{s}, \boldsymbol{\xi}^k)/\varsigma)^{\left(-\frac{\varsigma+1}{2}\right)} \tag{8.9}$$

This should avoid the *crowding problem* (van der Maaten and Hinton 2008), which may occur due to the volume difference between high-dimensional and low-dimensional spaces. The following formulas will give the most general definitions for flexible use of distances $d_\mathcal{X}$ and $d_\mathcal{E}$ and positive measures h and g in the high- and low-dimensional space, as well as explicit examples of them. Based on these settings, we define a novel cost function using the divergence $\mathrm{D}_{\mathrm{GKL}}$ Eq. (8.6):

$$E_{\mathrm{GKL}} \sim \int \sum_i \delta_{\Psi_{\mathrm{GKL}}(\boldsymbol{s}), \boldsymbol{x}^i} \cdot \sum_j \left[h_\sigma^{\Psi_{\mathrm{GKL}}(\boldsymbol{s})}(j) \log\left(\frac{h_\sigma^{\Psi_{\mathrm{GKL}}(\boldsymbol{s})}(j)}{g_\varsigma^{\boldsymbol{s}}(j)}\right) - h_\sigma^{\Psi_{\mathrm{GKL}}(\boldsymbol{s})}(j) + g_\varsigma^{\boldsymbol{s}}(j) \right] p(\boldsymbol{s})\, d\boldsymbol{s} \quad , \tag{8.10}$$

where the best match data point for $\boldsymbol{s}$ is defined as:

$$\Psi_{\mathrm{GKL}}(\boldsymbol{s}) = \boldsymbol{x}^i \text{ such that} \tag{8.11}$$
$$\sum_j \left[h_\sigma^{\Psi_{\mathrm{GKL}}(\boldsymbol{s})}(j) \log\left(\frac{h_\sigma^{\Psi_{\mathrm{GKL}}(\boldsymbol{s})}(j)}{g_\varsigma^{\boldsymbol{s}}(j)}\right) - h_\sigma^{\Psi_{\mathrm{GKL}}(\boldsymbol{s})}(j) + g_\varsigma^{\boldsymbol{s}}(j) \right] \text{ is minimum.}$$

The derivative of the cost function with respect to the images $\boldsymbol{\xi}^k$ yields the online learning update rule for a given sampling vector $\boldsymbol{s}$ (see 8.B for details):

$$\frac{\partial E_{\mathrm{GKL}}}{\partial \boldsymbol{\xi}^k} = \frac{\partial g_\varsigma^{\boldsymbol{s}}(k)}{\partial \boldsymbol{\xi}_k} \left(1 - \frac{h_\sigma^{\Psi_{\mathrm{GKL}}(\boldsymbol{s})}(k)}{g_\varsigma^{\boldsymbol{s}}(k)}\right) \quad , \tag{8.12}$$

In case of a Gaussian $g_\varsigma^{\boldsymbol{s}}(k)$ Eq. (8.8) the derivative reads:

$$\frac{\partial E_{\mathrm{GKL}}}{\partial \boldsymbol{\xi}^k} = \frac{1}{2\varsigma^2}\left(h_\sigma^{\Psi_{\mathrm{GKL}}(\boldsymbol{s})}(k) - g_\varsigma^{\boldsymbol{s}}(k)\right)\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)}{\partial \boldsymbol{\xi}^k} \tag{8.13}$$

$$= \frac{\alpha_g}{2}\left(h_\sigma^{\Psi_{\mathrm{GKL}}(\boldsymbol{s})}(k) - g_\varsigma^{\boldsymbol{s}}(k)\right)\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)}{\partial \boldsymbol{\xi}^k} \quad , \quad \alpha_g = \frac{1}{\varsigma^2} \tag{8.14}$$

and with a t-distributed $g_\varsigma^{\boldsymbol{s}}(k)$ defined in Eq. (8.9) the update is:

$$\frac{\partial E_{\mathrm{GKL}}}{\partial \boldsymbol{\xi}^k} = \frac{\varsigma+1}{2\varsigma}\frac{1}{(1+d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)/\varsigma)}\left(h_\sigma^{\Psi_{\mathrm{GKL}}(\boldsymbol{s})}(k) - g_\varsigma^{\boldsymbol{s}}(k)\right)\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)}{\partial \boldsymbol{\xi}^k} \tag{8.15}$$

$$= \frac{\alpha_t}{2}\left(h_\sigma^{\Psi_{\mathrm{GKL}}(\boldsymbol{s})}(k) - g_\varsigma^{\boldsymbol{s}}(k)\right)\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)}{\partial \boldsymbol{\xi}^k}, \quad \alpha_t = \frac{\varsigma+1}{\varsigma + d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)} \tag{8.16}$$

and we refer to this variant as t-distributed SONE (t-SONE).

Algorithm 8.2 : Self Organized Neighbor Embedding (SONE)

1: choose a structure hypothesis, given by sampling vectors $\boldsymbol{s} \in \mathcal{E}$
2: initialize the image vectors $\boldsymbol{\xi}$, e.g. randomly or by means of a PCA.
3: compute the neighborhood function, e.g. $h_\sigma^{\boldsymbol{x}^i}(j)$ Eq. (8.7)
4: **while** stopping criterion not reached **do**
5: present a sampling vector $\boldsymbol{s}$ from the structure hypothesis
6: compute the neighborhood cooperation $g_\varsigma^{\boldsymbol{s}}(k)$ in $\mathcal{E}$ $\forall \boldsymbol{\xi}^k$
7: find the best matching input vector $\Psi_{\mathrm{GKL}}(\boldsymbol{s})$ following Eq. (8.11)
8: perform the update of all image vectors $\boldsymbol{\xi}^k \leftarrow \boldsymbol{\xi}^k - \tau \cdot \frac{\partial E_{\mathrm{GKL}}}{\partial \boldsymbol{\xi}^k}$ following Eq. (8.13) or Eq. (8.15) dependent on the function $g_\varsigma^{\boldsymbol{s}}(k)$
9: **end while**

While the original XOM approach is based on attraction forces only (see Eq. (8.2)), the prototype update in Eq. (8.12) includes repulsion as well. This is due to the possibility of a change of the sign dependent on the fraction between the cooperativity function h and g. The XOM update emphasizes attraction and predominantly optimizes "continuity", such that small distances in $\mathcal{X}$ lead to small distances in $\mathcal{E}$. In contrast to the XOM adaptation rule, the SONE adaptation is able to push less similar samples out of a region of a sampling vector, if the pulling force of the actual winning sample is weaker than the repulsive force of the sampling vector. This also prevents image vectors of collapsing onto one point which is stated to be a problem in Locally Linear Embedding (LLE) (van der Maaten et al. 2009). Furthermore the parameter ς in the t-distributed version Eq. (8.15) can be used to control the granularity of the final embedding. Further information about the parameters can be found in section 8.4.

8.3.1 SONE without structure hypothesis

It is also possible to use this algorithm without a defined structure. One could simply change the definition of the sampling vectors, as inspired by (Wismüller 2001, Lee et al. 2003), in such a way that they are selected in close proximity to the image vector positions.

Therefore, instead of choosing a sampling vector randomly according to a given distribution, we visit the images $\boldsymbol{\xi}$ sequentially and choose a sampling vector $\boldsymbol{s}^j = \tilde{\boldsymbol{\xi}}^j$ drawn from a distribution centered around the actual images $\boldsymbol{\xi}^j$. Examples could be a Gaussian, a localized uniform, or a t-distribution. In our experiments we denote the use of this variant with the term *(ws)* added to the method name. And we used a normal distribution with variance ϖ: $\mathcal{N}(\boldsymbol{\xi}^j, \varpi)$. The algorithm thus changes to:

Algorithm 8.3 : SONE without structure hypothesis

1: initialize the image vectors $\boldsymbol{\xi}$, e.g. randomly or by means of a PCA.
2: compute the neighborhood function, e.g. $h_\sigma^{\boldsymbol{x}^i}(j)$ Eq. (8.7)
3: **while** stopping criterion not reached **do**
4: randomly pick an image vector $\boldsymbol{\xi}^j$
5: find a sampling vector drawn from $\mathcal{N}(\boldsymbol{\xi}^j, \varpi)$ centered around $\boldsymbol{\xi}^j$
6: compute the neighborhood cooperation $g_\zeta^{\boldsymbol{s}}(k)$ in $\mathcal{E}$ $\forall \boldsymbol{\xi}^k$
7: find the best matching input vector $\Psi_{\mathrm{GKL}}(\boldsymbol{s})$ following Eq. (8.11)
8: perform the update of all image vectors $\boldsymbol{\xi}^k \leftarrow \boldsymbol{\xi}^k - \tau \cdot \frac{\partial E_{\mathrm{GKL}}}{\partial \boldsymbol{\xi}^k}$
9: **end while**

The final positions of the vectors $\boldsymbol{\xi}$ represent the output of the algorithm. However, in this variant the SONE is not longer bounded to a predefined structure, but creates its own similarity map. Note, that in this variant the parameters have to be tuned carefully, so that the repulsive forces do not dominate the embedding. Furthermore the algorithm without structure hypothesis may be computationally more expensive if the number of data samples grows over the number of vectors, which would be used in a predefined structure.

8.4 Parameter setting

In this Section we will shortly discuss the parameters and their influence on the final embedding of the SONE algorithm. First, the dissimilarity measures $d_{\mathcal{X}}$ and $d_{\mathcal{E}}$ of the observation and embedding space have to be chosen. In our experiments we used the squared Euclidean distance for both of them. Further, one has to decide which neighborhood function g should be used in the embedding space. We show

in this section the different behavior of the algorithm for two example cases: Gaussian and t-distribution. As in XOM, the sampling vectors $\boldsymbol{s}$ may be chosen to match application-specific user needs. They could for example be drawn from a uniform distribution, a Gaussian, several Gaussian clusters or they could build a regular grid of any shape. In our experiments we used triangular grids generated by DISTMESH (Persson and Strang 2004). The list of parameters, which are candidates for adaptation during training, contains:

σ the variance of the neighborhood cooperation h in the observation space $\mathcal{X}$,
ς the variance of the neighborhood cooperation g in the embedding space $\mathcal{E}$,
τ the learning rate in the gradient decent optimization.

The parameter σ resembles the variance of the neighborhood function from the original SOM and XOM algorithms and is decreased during training. In our experiments, we used a different σ_i for every data sample $\boldsymbol{x}^i$ such that an ϵ-ball of variance σ_i would contain a fixed number n_k of neighbors. This ensures, that also data samples in less dense regions have an effect on the embedding. All σ_i follow an annealing scheme of the n_k during training:

$$n_k(t) = n_k(t_1) \cdot \exp\left(-\frac{\log\left(\frac{n_k(t_1)}{n_k(t_{\text{end}})}\right) t}{t_{\max}}\right), \tag{8.17}$$

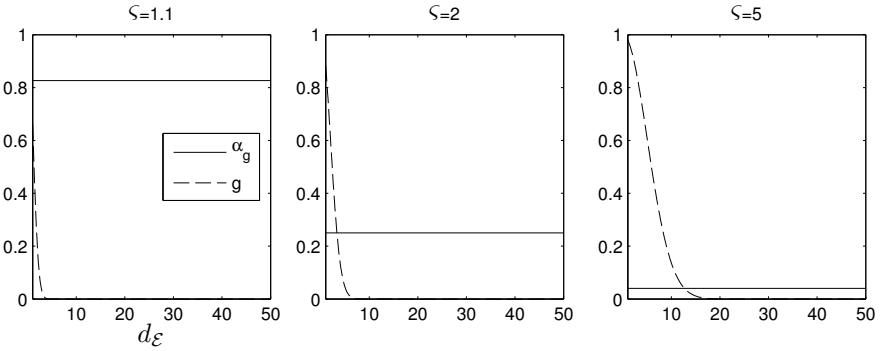
with $n_k(t_1)$ and $n_k(t_{\text{end}})$ being the number of neighbors at the beginning and at the end of training and $t_{\max}$ the total number of epochs (sweeps through the sampling vectors or number of iterations for randomly chosen $\boldsymbol{s}$). It is also possible to find appropriate σ_i by using the "perplexity" proposed for the SNE approach (Hinton and Roweis 2003).

From Eq. (8.3) follows that the winner always gets the maximal attraction force of one. Therefore, it is quite possible that for a sampling vector always the same data point $\boldsymbol{x}^i$ becomes the winner. To increase the probability that different samples become the winners to one sampling vector we adjusted the value of

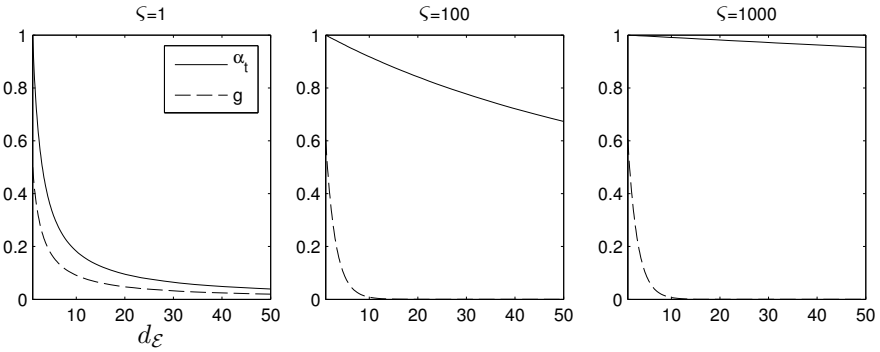
$$h_{\sigma_i}^{\boldsymbol{x}^i}(i) \leftarrow 0.9 \cdot \max_{i \neq j}\left(h_{\sigma_i}^{\boldsymbol{x}^i}(j)\right) \quad . \tag{8.18}$$

This way different samples become winner and therefore more data points influence the final embedding.

Figure 8.1 shows the influence of the parameter ς on the repulsive forces g and the learning rate α_* in dependence of the distance between image and sampling vectors in the embedding space. Fig. 8.1(a) shows the influence of the value ς for the repulsive forces addressed by g and the learning rate factor α_g in case of a Gaussian used as neighborhood function in the embedding space. The repulsion forces



(a) Gaussian neighborhood cooperation function in the embedding space



(b) t-distributed neighborhood cooperation function in the embedding space

Figure 8.1: Influence of the parameter ς on the repulsion forces g and the learning rate factor α_* in SONE for given distances $d_{\mathcal{E}}$. In (a) the neighborhood function g is Gaussian and α_g the resulting factor, see Eq. (8.14), which influences the learning rate τ. In (b) g is given by Eq. (8.9), α_t is defined in Eq. (8.16).

which may cause instabilities can be easily suppressed by big distances between the sampling vectors and a small $\varsigma \in [1, 2]$. For bigger ς, the update would become vanishingly small. In this case, the ς can be fixed during training, while the learning rate τ is decreased following an annealing scheme. One may also start with high repulsive forces denoted by a bigger value of ς and decrease it during training following an annealing scheme:

$$\varsigma(t) = \varsigma(t_1) \cdot \exp\left(-\frac{\log\left(\frac{\varsigma(t_1)}{\varsigma(t_{\text{end}})}\right) t}{t_{\max}}\right), \tag{8.19}$$

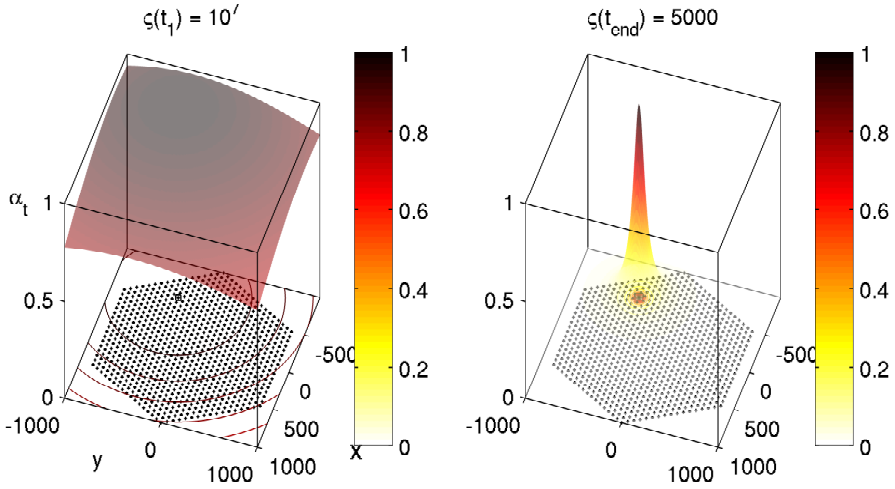


Figure 8.2: The influence of the parameter ς for the learning rate factor α_t in t-SONE using a t-distribution in the embedding space. The sampling vectors lie on a regular grid of hexagonal shape. For big values of ς, all image vectors are updated with nearly equal strength. With smaller values the update strength of image vectors outside the direct neighborhood of a sampling vector is suppressed.

with $\varsigma(t_1)$ and $\varsigma(t_{\text{end}})$ being the value of ς in the beginning and the end of the training. Note that in this case the learning rate τ should be adapted inversely proportional to the factor α_g, so that the resulting learning rate factor $\tau \cdot \alpha_g$ is decreased during training.

The application of a t-distribution in the embedding space shows an interesting behavior of the update strength α_t in dependence of the distance $d_{\mathcal{E}}(\boldsymbol{s}, \boldsymbol{\xi}^j)$ (see Fig. 8.1(b)). Here, the localization of the update in the embedding space can be controlled with ς. A high value of ς ensures the same update strength for all samples. For lower values only samples in the direct neighborhood of the actual sampling vector are updated, see Fig. 8.2. With the parameter ς for the t-distribution we can control the granularity or level of detail in the final similarity map. The influence of the learning rate τ is negligible in this case and it is fixed to one. The value of ς is decreased during training with a similar annealing scheme as Eq. (8.19).

In summary, the parameter which depends on the actual data set at hand is σ for the neighborhood function in the observation space $\mathcal{X}$. The other parameters like the sampling distribution $\boldsymbol{s}$ are dependent on the needs and preferences of the user, but not on the data itself. As in original XOM, prior knowledge may be integrated in the choice of the structure. The parameter ς for the cooperativity function in the

embedding space is adjusted according to the choice of the structure hypothesis and the level of detail the user desires.

8.5 Complexity

The complexity of the structure variant of SONE depends on the dimension M of the embedding space $\mathcal{E}$, the number of samples to embed n, the number of sampling vectors $n_{\boldsymbol{s}}$ (which is usually much smaller than n) and the number of epochs $t_{\max}$. So, every epoch calculations of the complexity $\mathcal{O}(M \cdot n \cdot n_{\boldsymbol{s}})$ have to be computed.

Fig. 8.3 shows the computational advantage of the simplest variant of SONE in dependence of the number of data points to be embedded. For SNE and SONE we used the same number of 1000 iterations and run the simulation on the same machine and all of them were matlab implementations. Most of the proposed dimension reduction techniques show at least quadratic complexity with the number of points to process. In those methods, the computation of the pairwise distances of the image vectors is necessary in every iteration. The structure variant of SONE on the other hand only requires the computation of the distance of the image vectors to a given sampling vector in each iteration. Thus, for a sweep through the sampling set (one epoch) the complexity is dependent on the number of sampling vectors and the number of points, which is less then quadratic, if the number of sampling vectors is smaller than the data set size.

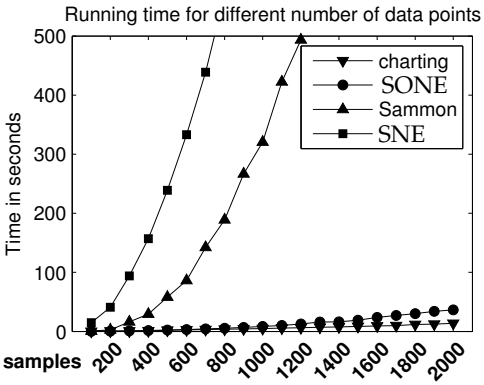


Figure 8.3: The running time of different dimension reduction methods depending on the number of samples to embed.

8.6 Experiments

In this Section we show the results of different versions of two-dimensional SONE on three exemplary real world data sets. We compare some conventional quality measures, like the Sammon's stress (Sammon 1969), Spearmans and Pearsons correlation (ρ_s and ρ_p) (Venna 2007) as well as the Nearest Neighbor (1-NN) Error ($\mathcal{E}_{NN}$) and the Intrusion / Extrusion measure (see Algorithm 6.6) proposed by (Lee and

Verleysen 2008, Lee and Verleysen 2009), on the embeddings. Some methods we compare display linear complexity with the number of points, namely PCA and charting (Lee and Verleysen 2007). Additionally, we compare the results to those obtained from t-SNE, which is widely accepted as a high quality state-of-the-art technique, although it exhibits higher complexity and is computationally more expensive than the other techniques.

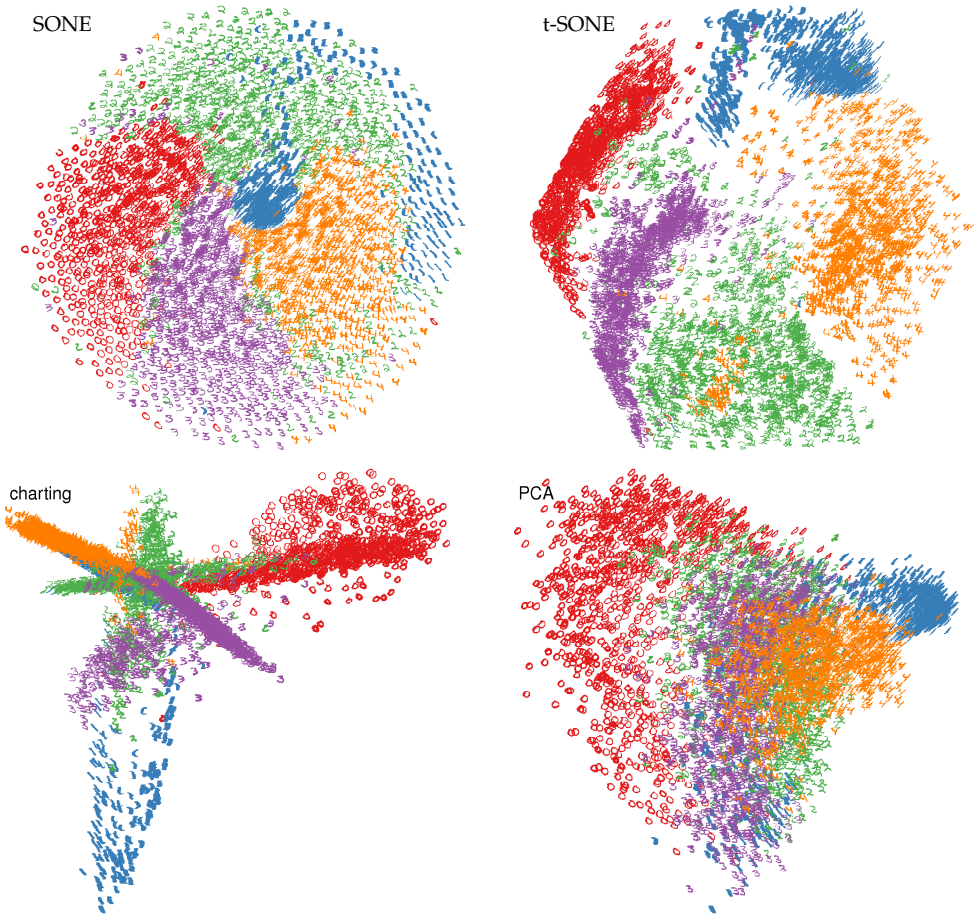


Figure 8.4: Example embeddings of the USPS Digits data set. From the upper left till lower right it shows: First, one result for the SONE with Gaussian g and sampling vectors forming a regular circle ($\mathcal{E}_{NN} = 0.13$), second, one results of t-SONE using a regular sampling grid of hexagonal structure ($\mathcal{E}_{NN} = 0.05$), third, an example result of charting with 6 analyzers ($\mathcal{E}_{NN} = 0.26$), and last, the result of PCA ($\mathcal{E}_{NN} = 0.37$).

8.6.1 USPS digits

The USPS Digits dataset from the UCI repository (Asuncion et al. 1998) consists of images of hand-written digits as already explained in previous Chapters. For clarity, we use the digits $\in \{0, 1, 2, 3, 4\}$, resulting in 5500 samples. The parameter settings of all reduction techniques were optimized for performance, and on each parameter we performed 10 independent runs. For charting and t-SNE, we used the code provided by (van der Maaten et al. 2009). Charting yielded reasonable results for six analysers, while for t-SNE a perplexity of 45 provided good results. The other parameters were chosen according to default values provided by (van der Maaten et al. 2009). Some example embeddings are shown in Fig. 8.4 and the quality with different measures is shown in Fig. 8.5 and Table 8.1. The results of the SONE algorithm were investigated using different variants: with and without structure hypothesis and with Gaussian and t-distribution in the embedding space respectively. The parameter settings can be found in table 8.2 on page 165.

Table 8.1: Quality measures for USPS.

Method	t-SONE	charting	t-SNE
Sammon	**0.16** (0.0)	0.25 (0.1)	**0.16** (0.0)
ρ_s	**0.54** (0.0)	0.42 (0.1)	0.40 (0.1)
ρ_p	**0.57** (0.0)	0.43 (0.1)	0.44 (0.1)
$\mathcal{E}_{NN}$	0.06 (0.0)	0.29 (0.1)	**0.02** (0.0)

The top left panel in Fig. 8.4 shows an example embedding of the SONE algorithm with a Gaussian neighborhood function in the embedding space. In the top right panel an example embedding of the t-SONE algorithm using a t-distribution in the embedding space is presented. Table 8.1 shows the results for the Sammon's stress, Spearmans and Pearsons correlation (ρ_s and ρ_p) for the different dimension reduction methods and the t-SONE with structure using t-distribution. Two example results for embeddings without a structure hypothesis are shown in Figure 8.6. The left side was achieved with SONE(ws) using a Gaussian neighborhood and the right side is an example result of t-SONE(ws).

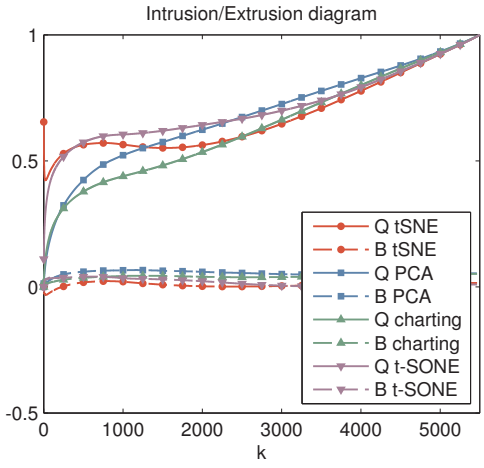


Figure 8.5: Values of the overall quality Q and B versus the number of neighbors k.

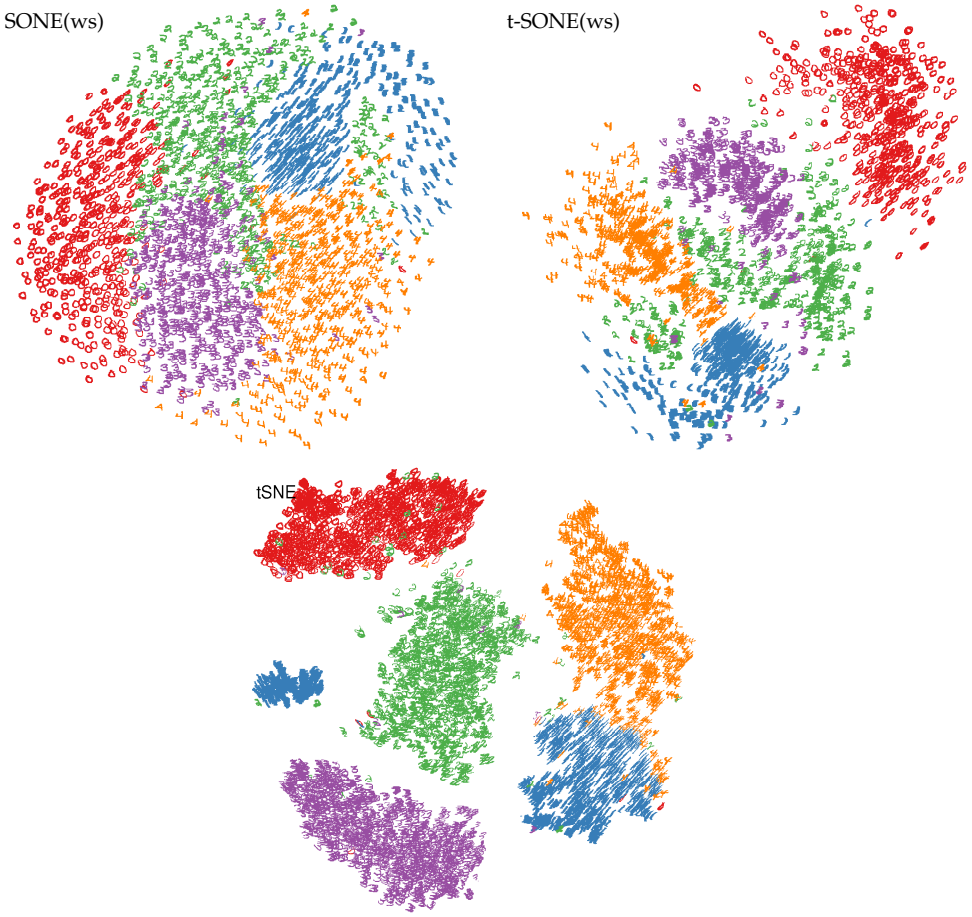


Figure 8.6: Two example embeddings of the SONE algorithm without a structure hypothesis and an t-SNE example embedding. For the left- and right-hand side, a Gaussian and a t-distribution was used in the embedding space, respectively.

From Fig. 8.5 and Table 8.1 can be reasoned that the t-SONE embedding can be identified as a competitive trade-off between high embedding quality and low computational expense. The different variants result in different behavior of the embeddings: the incorporation of a Gaussian in the embedding space leads to similarity maps which preserve local neighborhoods, but prevents the image vectors of being projected onto each other. In addition, it forces image vectors to fill the whole structure. Using the t-distributed variant, the t-SONE shows the ability of creating gaps between classes, and, using a small ς the image vectors are not forced to

spread in empty regions of the sampling space. In contrast to t-SNE (see Fig. 8.6) the (t-)SONE embeddings with structure hypothesis (see Fig. 8.4) represent the different variances of the classes presented by the space they occupy in the embeddings. The digits equal to one are always confined to a small number of sampling vectors, whereas the twos and fours occupy a big region.

8.6.2 Relational data

As the SONE algorithm depends on the topology of the observed data only, it can deal with pairwise distances as input. This is a property that SONE directly inherits from the original XOM algorithm, which has been applied to the visualization of non-metric real-world data. These data sets are known as dissimilarity or relational data sets and they are often found in biological real world problems, in which a data representation in vector form is not feasible.

As two examples we chose the Cat Cortex data set (Graepel et al. 1999) preprocessed by Haasdonk (Haasdonk and Bahlmann 2004) and the Protein data set (Mevissen and Vingron 1996). The Cat Cortex originates from anatomic studies of cats' brains. This data set is given as a matrix containing the connection strength between 65 cortical areas spit into four classes corresponding to four different regions of the cortex. The similarity matrix is symmetric but the triangle inequality does not hold. The Protein data contains the evolutionary distances of 226 globin proteins (Mevissen and Vingron 1996). We use the five classes proposed in (Haasdonk and Bahlmann 2004): HA, HB, MY, GG/GP and others. The class others combines small classes form the original dataset and represents only a small fraction of the whole data set.

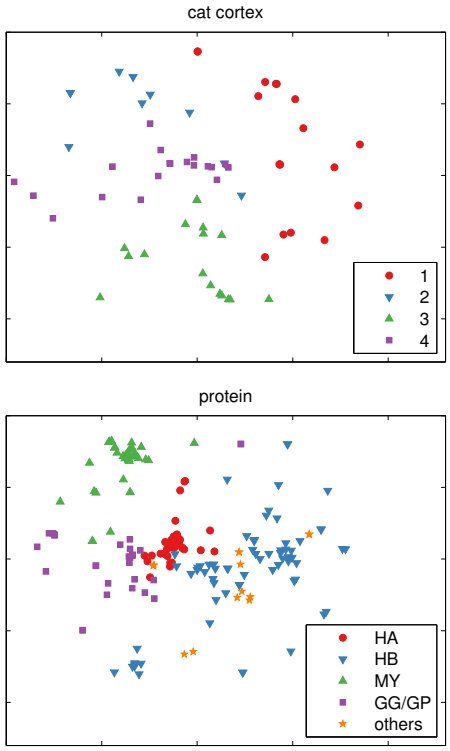


Figure 8.7: Embeddings of Cat Cortex ($\mathcal{E}_{NN} = 0.09$) and Protein ($\mathcal{E}_{NN} = 0.04$).

Fig. 8.7 shows two example embeddings of the relational data sets. We run the t-SONE algorithm 10 times for each

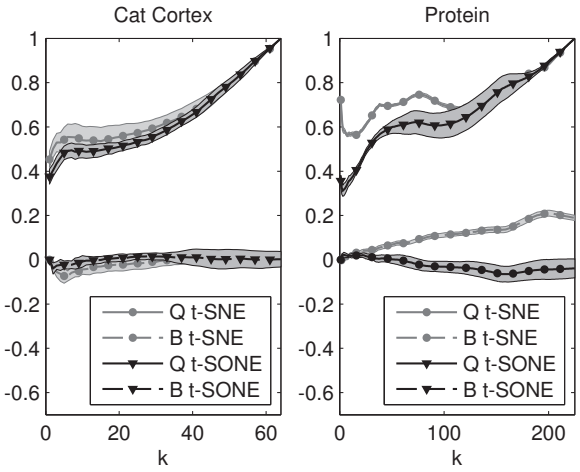


Figure 8.8: The overall embedding quality Q and B for two relational data sets. The gray shaded area denotes the STD.

data set with random initialization of the image vectors. The embedding quality is measured by Q and the behavior with B and compared to those from t-SNE with varying perplexity. The mean values and standard deviation (STD) of these measures is shown in Fig. 8.8. For t-SNE the best results were achieved with perplexity 25. The parameter setting for the t-SONE for the Cat Cortex and for the Protein data can be found in table 8.2 on page 165.

The quality of the embeddings of t-SONE and t-SNE is comparable. With the Cat Cortex data t-SNE shows bigger standard deviation regarding the random initialization and more extrusive behavior for small neighborhoods. For the Protein data the quality measured by Q is higher with t-SNE and the embedding shows highly intrusive behavior. The t-SONE embedding shows in this case extrusive behavior. This shows, that despite the close relationship of SNE and SONE even the behavior of the embeddings may vary a lot. The mean 1-NN Error of the 10 t-SONE embeddings is $\bar{\mathcal{E}}_{NN} = 0.13$ with standard deviation of 3% for the Cat Cortex and $\bar{\mathcal{E}}_{NN} = 0.08$ with STD=3% for the Protein data set.

8.7 Conclusion

In this contribution, we have introduced an extension of the XOM for structure-preserving dimensionality reduction. Based on minimizing the Kullback-Leibler divergence of neighborhood functions in data and embedding space, SONE creates

Table 8.2: Explicit parameter settings for the SONE variants in the experiments.

method	structure hypothesis	$t_{\max}$	σ_i	ς Eq. (8.19)
USPS				
SONE	triangular mesh, in form of a circle, 562 $\boldsymbol{s}$	50	$\sigma_i(t_1)$ =perplexity 30 $\sigma_i(t_{\text{end}})$ =perplexity 3	$\varsigma = 1$
t-SONE	triangular mesh, in form of a hexagon	500	Eq. (8.17), $n_k(t_1) = 3000$ $n_k(t_{\text{end}}) = 10$	$\varsigma(t_1) = 10^7$ $\varsigma(t_{\text{end}}) = 5000$
SONE(ws)	no hypothesis! $\boldsymbol{s}$ drawn from $\mathcal{N}(\boldsymbol{\xi}^j, 0.1)$	300	$\sigma_i(t_1)$ =perplexity 500 $\sigma_i(t_{\text{end}})$ =perplexity 5	$\varsigma = 1$
t-SONE(ws)	no hypothesis! $\boldsymbol{s}$ drawn from $\mathcal{N}(\boldsymbol{\xi}^j, 10)$	300	$\sigma_i(t_1)$ =perplexity 500 $\sigma_i(t_{\text{end}})$ =perplexity 5	$\varsigma(t_1) = 10^7$ $\varsigma(t_{\text{end}}) = 0.1$
Cat Cortex				
t-SONE	triangular mesh, in form of a hexagon, 48 $\boldsymbol{s}$	500	Eq. (8.17), $n_k(t_1) = 50$ $n_k(t_{\text{end}}) = 5$	$\varsigma(t_1) = 10^7$ $\varsigma(t_{\text{end}}) = 1000$
Protein				
t-SONE	triangular mesh, in form of a hexagon, 200 $\boldsymbol{s}$	500	Eq. (8.17), $n_k(t_1) = 200$ $n_k(t_{\text{end}}) = 5$	$\varsigma(t_1) = 10^7$ $\varsigma(t_{\text{end}}) = 2000$

a conceptual link between fast sequential online learning known from topology-preserving mappings and principled direct divergence optimization approaches, such as SNE and t-SNE. Quantitative comparative evaluation on benchmark data using multiple embedding quality measures identifies SONE as a competitive trade-off between high embedding quality and low computational expense, which motivates its extended use in real-world settings throughout science and engineering. We have extended the algorithm to utilize different distributions, namely the Gaussian and the t-distribution following the ideas proposed in t-SNE (van der Maaten and Hinton 2008). We have analyzed different variants of the SONE algorithm with and without structure hypothesis and using different distributions, which offers high flexibility based on application needs. Finally, it allows the user to incorporate prior knowledge and the tuning of the level of detail the user desires. The extension of this algorithm to arbitrary divergences will be addressed in the next Chapter.

8.A Derivative of the XOM cost function

We write the derivative of the cost function Eq. (8.5) with respect to $\boldsymbol{\xi}^k$:

$$\frac{\partial E_{\mathrm{XOM}}}{\partial \boldsymbol{\xi}^k} = \int \sum_i \frac{\partial \delta_{\Psi(\boldsymbol{s}),\boldsymbol{x}^i}}{\partial \boldsymbol{\xi}^k} \sum_j h_\sigma^{\boldsymbol{x}^i}(j) \cdot d_{\mathcal{E}}(\boldsymbol{s}, \boldsymbol{\xi}^j)\, p(\boldsymbol{s}) d\boldsymbol{s} + \int h_\sigma^{\Psi(\boldsymbol{s})}(k) \frac{\partial d_{\mathcal{E}}(\boldsymbol{s}, \boldsymbol{\xi}^k)}{\partial \boldsymbol{\xi}^k}\, p(\boldsymbol{s}) d\boldsymbol{s}. \quad (8.20)$$

The second term yields the learning rule Eq. (8.2) while the first term vanishes due to the following considerations: We use the shorthand notation

$$\Phi(\boldsymbol{x}^i, \boldsymbol{s}) = \sum_j h_\sigma^{\boldsymbol{x}^i}(j) \cdot d_{\mathcal{E}}(\boldsymbol{s}, \boldsymbol{\xi}^j) \ . \quad (8.21)$$

Then, the Kronecker delta can be expressed as

$$\delta_{\Psi(\boldsymbol{s}),\boldsymbol{x}^i} = H\Big(\sum_k H(\Phi(\boldsymbol{x}^i, \boldsymbol{s}) - \Phi(\boldsymbol{x}^k, \boldsymbol{s})) - n + 0.5 \Big) \ , \quad (8.22)$$

where H denotes the Heaviside function and n denotes the number of data points $\boldsymbol{x}^i$. The derivative of H is given by the delta function δ which is symmetric and non-vanishing only for input zero. Hence the first term of Eq. (8.20) vanishes:

$$\begin{aligned} &\int \left[\sum_i \frac{\partial \delta_{\Psi(\boldsymbol{s}),\boldsymbol{x}^i}}{\partial \boldsymbol{\xi}^k} \Phi(\boldsymbol{x}^i, \boldsymbol{s}) \right] p(\boldsymbol{s}) d\boldsymbol{s} \\ &= \int \sum_i \delta \left(\sum_l H(\Phi(\boldsymbol{x}^i, \boldsymbol{s}) - \Phi(\boldsymbol{x}^l, \boldsymbol{s})) - n + 0.5 \right) \cdot \sum_l \delta(\Phi(\boldsymbol{x}^i, \boldsymbol{s}) - \Phi(\boldsymbol{x}^l, \boldsymbol{s})) \cdot \\ &\qquad \Big(h_\sigma^{\boldsymbol{x}^i}(k) - h_\sigma^{\boldsymbol{x}^l}(k) \Big) \cdot \frac{\partial d_{\mathcal{E}}(\boldsymbol{s}, \boldsymbol{x}^k)}{\partial \boldsymbol{\xi}^k} \sum_j h_\sigma^{\boldsymbol{x}^i}(j) \cdot d_{\mathcal{E}}(\boldsymbol{s}, \boldsymbol{\xi}^j)\, p(\boldsymbol{s}) d\boldsymbol{s} \quad (8.23) \end{aligned}$$

$$\begin{aligned} &= \int \Bigg[\sum_{ilj} \delta \left(\sum_{l'} H(\Phi(\boldsymbol{x}^i, \boldsymbol{s}) - \Phi(\boldsymbol{x}^{l'}, \boldsymbol{s})) - n + \frac{1}{2} \right) \cdot \delta(\Phi(\boldsymbol{x}^i, \boldsymbol{s}) - \Phi(\boldsymbol{x}^l, \boldsymbol{s})) \cdot \\ &h_\sigma^{\boldsymbol{x}^i}(k) \cdot h_\sigma^{\boldsymbol{x}^i}(j) \cdot d_{\mathcal{E}}(\boldsymbol{s}, \boldsymbol{\xi}^j) - \sum_{ilj} \delta \left(\sum_{l'} H(\Phi(\boldsymbol{x}^l, \boldsymbol{s}) - \Phi(\boldsymbol{x}^{l'}, \boldsymbol{s})) - n + \frac{1}{2} \right) \\ &\cdot \delta(\Phi(\boldsymbol{x}^l, \boldsymbol{s}) - \Phi(\boldsymbol{x}^i, \boldsymbol{s})) \cdot h_\sigma^{\boldsymbol{x}^i}(k) \cdot h_\sigma^{\boldsymbol{x}^l}(j) \cdot d_{\mathcal{E}}(\boldsymbol{s}, \boldsymbol{\xi}^j) \Bigg] \cdot \frac{\partial d_{\mathcal{E}}(\boldsymbol{s}, \boldsymbol{x}^k)}{\partial \boldsymbol{\xi}^k}\, p(\boldsymbol{s}) d\boldsymbol{s} \quad (8.24) \end{aligned}$$

$$= \int \Bigg[\sum_{il} \delta \left(\sum_{l'} H(\Phi(\boldsymbol{x}^i, \boldsymbol{s}) - \Phi(\boldsymbol{x}^{l'}, \boldsymbol{s})) - n + \frac{1}{2} \right) \cdot \delta(\Phi(\boldsymbol{x}^i, \boldsymbol{s}) - \Phi(\boldsymbol{x}^l, \boldsymbol{s})) \cdot$$
$$h_\sigma^{\boldsymbol{x}^i}(k) \cdot \Phi(\boldsymbol{x}^i, \boldsymbol{s}) - \sum_{il} \delta \left(\sum_{l'} H(\Phi(\boldsymbol{x}^l, \boldsymbol{s}) - \Phi(\boldsymbol{x}^{l'}, \boldsymbol{s})) - n + \frac{1}{2} \right)$$
$$\cdot \delta(\Phi(\boldsymbol{x}^i, \boldsymbol{s}) - \Phi(\boldsymbol{x}^l, \boldsymbol{s})) \cdot h_\sigma^{\boldsymbol{x}^i}(k) \cdot \Phi(\boldsymbol{x}^l, \boldsymbol{s}) \Bigg] \cdot \frac{\partial d_{\mathcal{E}}(\boldsymbol{s}, \boldsymbol{\xi}^k)}{\partial \boldsymbol{\xi}^k} \, p(\boldsymbol{s}) d\boldsymbol{s} \quad = 0. \quad (8.25)$$

8.B Derivative of the SONE cost function

The derivatives of the neighborhood function read in case of a Gaussian Eq. (8.8):

$$\frac{\partial g_\varsigma^{\boldsymbol{s}}(k)}{\partial \boldsymbol{\xi}^k} = \left(-\frac{g_\varsigma^{\boldsymbol{s}}(k)}{2\varsigma^2} \right) \frac{\partial d_{\mathcal{E}}(\boldsymbol{s}, \boldsymbol{\xi}^k)}{\partial \boldsymbol{\xi}^k} \quad (8.26)$$

and in case of a t-distribution Eq. (8.9)

$$\frac{\partial g_\varsigma^{\boldsymbol{s}}(k)}{\partial \boldsymbol{\xi}^k} = \left(-\frac{\varsigma + 1}{2\varsigma} \right) \frac{g_\varsigma^{\boldsymbol{s}}(k)}{(1 + d_{\mathcal{E}}(\boldsymbol{s}, \boldsymbol{\xi}^k)/\varsigma)} \frac{\partial d_{\mathcal{E}}(\boldsymbol{s}, \boldsymbol{\xi}^k)}{\partial \boldsymbol{\xi}^k} \quad . \quad (8.27)$$

We write the derivative of the cost function Eq. (8.10) with respect to $\boldsymbol{\xi}^k$:

$$\frac{\partial E_{\mathrm{GKL}}}{\partial \boldsymbol{\xi}^k} \sim \int \sum_i \frac{\partial \delta_{\Psi_{\mathrm{GKL}}(\boldsymbol{s}), \boldsymbol{x}^i}}{\partial \boldsymbol{\xi}^k} \sum_j \left(h_\sigma^{\boldsymbol{x}^i}(j) \log \left(\frac{h_\sigma^{\boldsymbol{x}^i}(j)}{g_\varsigma^{\boldsymbol{s}}(j)} \right) - h_\sigma^{\boldsymbol{x}^i}(j) + g_\varsigma^{\boldsymbol{s}}(j) \right) p(\boldsymbol{s}) d\boldsymbol{s}$$
$$+ \int \sum_i \delta_{\Psi_{\mathrm{GKL}}(\boldsymbol{s}), \boldsymbol{x}^i} \cdot \frac{\partial g_\varsigma^{\boldsymbol{x}}(k)}{\partial \boldsymbol{\xi}^k} \left(1 - \frac{h_\sigma^{\boldsymbol{x}^i}(k)}{g_\varsigma^{\boldsymbol{s}}(k)} \right) p(\boldsymbol{s}) d\boldsymbol{s} \quad , \quad (8.28)$$

with $\Psi_{\mathrm{GKL}}(\boldsymbol{s})$ defined in Eq. (8.11). The latter term yields the learning rule. The first term vanishes, as can be seen as follows: We use the shorthand notation

$$\Phi^{\mathrm{N}}(\boldsymbol{x}^i, \boldsymbol{s}) = \sum_j \left(h_\sigma^{\boldsymbol{x}^i}(j) \log \left(\frac{h_\sigma^{\boldsymbol{x}^i}(j)}{g_\varsigma^{\boldsymbol{s}}(j)} \right) - h_\sigma^{\boldsymbol{x}^i}(j) + g_\varsigma^{\boldsymbol{s}}(j) \right) \quad . \quad (8.29)$$

Then the best match input point can be expressed as

$$\delta_{\Psi_{\mathrm{GKL}}(\boldsymbol{s}), \boldsymbol{x}^i} = H\Big(\sum_k H(\Phi^{\mathrm{N}}(\boldsymbol{x}^i, \boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^k, \boldsymbol{s})) - n + \frac{1}{2} \Big) \quad . \quad (8.30)$$

Hence the additional first term of Eq. (8.28) vanishes, because of following:

$$\begin{aligned}
&\int \sum_i \frac{\partial \delta_{\Psi_{\mathrm{GKL}}(\boldsymbol{s}),\boldsymbol{x}^i}}{\partial \boldsymbol{\xi}^k} \cdot \Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s})\; p(\boldsymbol{s}) d\boldsymbol{s} \\
&= \int \sum_i \delta \left[\sum_l H(\Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^l,\boldsymbol{s})) - n + \frac{1}{2} \right] \sum_l \delta \left(\Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^l,\boldsymbol{s}) \right) \\
&\qquad \cdot \left[(g_\varsigma^{\boldsymbol{s}}(k) - h_\sigma^{\boldsymbol{x}^i}(k)) - (g_\varsigma^{\boldsymbol{s}}(k) - h_\sigma^{\boldsymbol{x}^l}(k)) \right] \frac{\partial g_\varsigma^{\boldsymbol{s}}(k)}{\partial \boldsymbol{\xi}^k} \qquad (8.31)
\end{aligned}$$

$$\begin{aligned}
&\sum_j \left(h_\sigma^{ij} \log \left(\frac{h_\sigma^{ij}}{g_\gamma^j} \right) - h_\sigma^{\boldsymbol{x}^i}(j) + g_\varsigma^{\boldsymbol{s}}(j) \right) \; p(\boldsymbol{s}) d\boldsymbol{s} \\
&= \int \sum_{ilj} \delta \left(\sum_{l'} H(\Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^{l'},\boldsymbol{s})) - n + \frac{1}{2} \right) \cdot \delta(\Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^l,\boldsymbol{s})) \\
&\cdot (g_\varsigma^{\boldsymbol{s}}(k) - h_\sigma^{\boldsymbol{x}^i}(k)) \frac{\partial g_\varsigma^{\boldsymbol{s}}(k)}{\partial \boldsymbol{\xi}^k} \cdot \left(h_\sigma^{\boldsymbol{x}^i}(j) \log \left(\frac{h_\sigma^{\boldsymbol{x}^i}(j)}{g_\varsigma^{\boldsymbol{s}}(j)} \right) - h_\sigma^{\boldsymbol{x}^i}(j) + g_\varsigma^{\boldsymbol{s}}(j) \right) \; p(\boldsymbol{s}) d\boldsymbol{s} \\
&- \int \sum_{ilj} \delta \left(\sum_{l'} H(\Phi^{\mathrm{N}}(\boldsymbol{x}^l,\boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^{l'},\boldsymbol{s})) - n + \frac{1}{2} \right) \cdot \delta(\Phi^{\mathrm{N}}(\boldsymbol{x}^l,\boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s})) \\
&\cdot (g_\varsigma^{\boldsymbol{s}}(k) - h_\sigma^{\boldsymbol{x}^i}(k)) \frac{\partial g_\varsigma^{\boldsymbol{x}}(k)}{\partial \boldsymbol{\xi}^k} \cdot \left(h_\sigma^{\boldsymbol{x}^l}(j) \log \left(\frac{h_\sigma^{\boldsymbol{x}^l}(j)}{g_\varsigma^{\boldsymbol{s}}(j)} \right) - h_\sigma^{\boldsymbol{x}^l}(j) + g_\varsigma^{\boldsymbol{s}}(j) \right) \; p(\boldsymbol{s}) d\boldsymbol{s}
\end{aligned} \qquad (8.32)$$

$$\begin{aligned}
&= \int \sum_{il} \delta \left(\sum_{l'} H(\Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^{l'},\boldsymbol{s})) - n + \frac{1}{2} \right) \cdot \delta(\Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^l,\boldsymbol{s})) \\
&\cdot (g_\varsigma^{\boldsymbol{s}}(k) - h_\sigma^{\boldsymbol{x}^i}(k)) \cdot \frac{\partial g_\varsigma^{\boldsymbol{s}}(k)}{\partial \boldsymbol{\xi}^k} \cdot \Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s})\; p(\boldsymbol{s}) d\boldsymbol{s} \\
&- \int \sum_{il} \delta \left(\sum_{l'} H(\Phi^{\mathrm{N}}(\boldsymbol{x}^l,\boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^{l'},\boldsymbol{s})) - n + \frac{1}{2} \right) \cdot \delta(\Phi^{\mathrm{N}}(\boldsymbol{x}^l,\boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s})) \\
&\qquad \cdot (g_\varsigma^{\boldsymbol{s}}(k) - h_\sigma^{\boldsymbol{x}^i}(k)) \cdot \frac{\partial g_\varsigma^{\boldsymbol{s}}(k)}{\partial \boldsymbol{\xi}^k} \cdot \Phi^{\mathrm{N}}(\boldsymbol{x}^l,\boldsymbol{s})\; p(\boldsymbol{s}) d\boldsymbol{s} \quad = 0, \qquad (8.33)
\end{aligned}$$

because of the symmetry of δ and the fact that δ is nonvanishing only if $\Phi^{\mathrm{N}}(\boldsymbol{x}^l,\boldsymbol{s}) = \Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s})$.

Published as:

K. Bunte, F.-M. Schleif, S. Haase and T. Villmann – *"Mathematical Foundations of the Self Organized Neighbor Embedding (SONE) for Dimension Reduction and Visualization,"* in Proc. of ESANN, pp. 29–34, Bruges, Belgium, April 2011.

K. Bunte, F.-M. Schleif, S. Haase and T. Villmann – *"Stochastic Neighbor Embedding (SNE) for Dimension Reduction and Visualization using arbitrary Divergences,"* submitted to Neurocomputing 2011.

Chapter 9

Non-linear Dimension Reduction Employing Divergences

The nice thing about standards is that there are so many of them to choose from.

Andrew S. Tanenbaum (Computer Networks, p.254)

Abstract

We present a systematic approach to the mathematical treatment of the Self Organized Neighbor Embedding, Stochastic Neighbor Embedding and t-distributed SNE. This allows an easy adaptation of the methods or exchange of their respective modules. In particular, the divergence which measures the difference between distributions in the original and the embedding space can be treated independently from other components like, e.g., the similarity of data points or the distribution. We focus on the extension for different divergences and propose a general framework based on the concept of Fréchet-derivatives. This way the general approach can be adapted to the user specific needs. We derive the explicit learning rules for a wide range of divergences and concentrate on the evaluation of the Gamma-divergence for t-distributed SNE and Self Organized Neighbor Embedding on several real-world data sets.

9.1 Introduction

Many dimension reduction methods have been introduced and discussed in the previous Chapters based on different objectives. Recently, the Stochastic Neighbor Embedding (SNE) (Hinton and Roweis 2003) and extensions thereof have become popular for visualization. It approximates the probability distribution

in the high-dimensional space, defined by neighboring points, with the corresponding probability distribution in a lower-dimensional space. In (van der Maaten and Hinton 2008) a technique called t-distributed SNE (t-SNE) is proposed, which is a variation of SNE considering a particular statistical model assumption for the low-dimensional distribution. The similarity of the distributions is quantified in terms of the Kullback-Leibler (KL) divergence. A computational efficient combination of fast sequential online learning and principled direct divergence optimization known from SNE is called Self Organized Neighbor Embedding (SONE) (Bunte, Hammer, Villmann, Biehl and Wismüller 2010), see Chapter 8. All these methods measure the disagreement of a topology defining functions in the high-dimensional space and the low-dimensional space by means of the KL or the generalized Kullback-Leibler (GKL) divergence. Functional metrics like Sobolev distances, kernel-based dissimilarity measures and divergences have attracted attention recently for the processing of data showing a functional structure. These metrics were for example investigated as alternatives to the most common choice, the Euclidean distance (Rossi et al. 2005, Lee and Verleysen 2005, Ramsay and Silverman 2006, Villmann 2007, Villmann and Schleif 2009). The application of divergences for Vector quantization and Learning Vector Quantization schemes have been investigated in (Villmann and Haase 2011, Mwebaze et al. 2011).

In this Chapter, we formulate a mathematical framework based on Fréchet derivatives which allows to generalize the concept of SONE, SNE and t-SNE to arbitrary divergences. This leads to a new dimension reduction and visualization scheme, which can be adapted to the user specific requirements in an actual problem. We summarize the general classes of divergences following the scheme introduced by (Cichocki et al. 2009) and extended in (Villmann and Haase 2011). The mathematical framework for functional derivatives of continuous divergences is given by the functional-analytic generalization of common derivatives, known as Fréchet derivatives (Frigyik et al. 2008, Kantorowitsch and Akilow 1978). It is the generalization of partial derivatives used for the discrete variants of the divergences. After characterizing the different classes of divergences and the introduction of Fréchet derivatives, we introduce a general mathematical view on the SONE, SNE and t-SNE algorithms incorporating these principles. Real world data sets demonstrate the applicability of this approach.

9.2 Specifications of divergences

Divergences are functionals $\mathrm{D}(p\|q)$ designed as dissimilarity measures between two nonnegative integrable functions p and q (Cichocki et al. 2009). In practice, usually

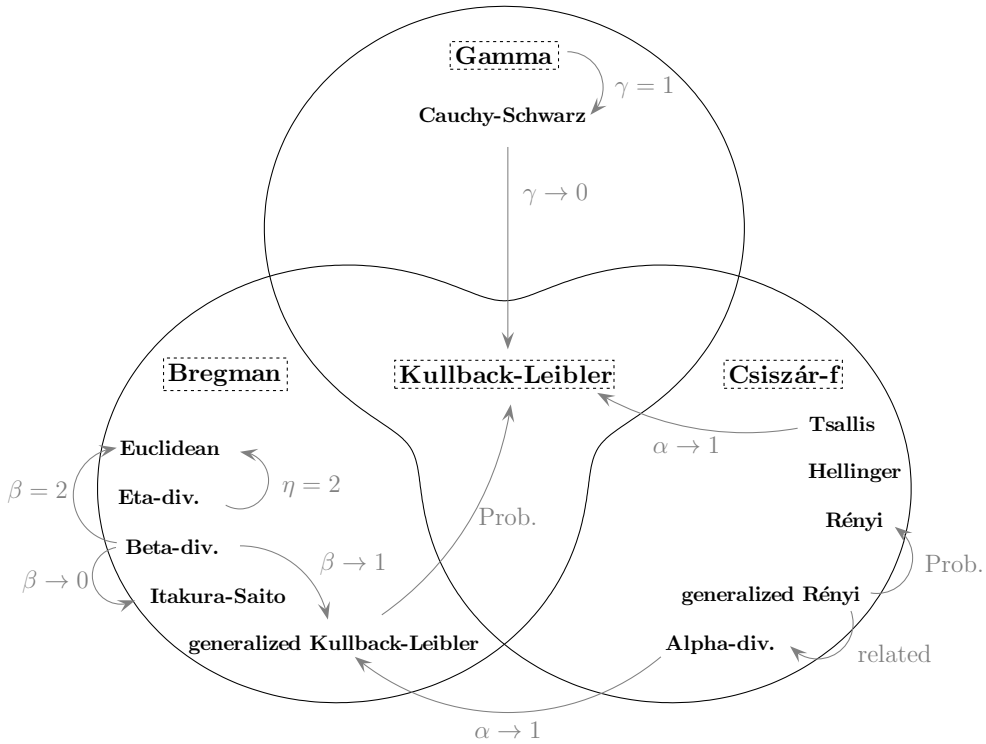


Figure 9.1: Overview over the families of divergences and their relationship to each other. The shortcut *Prob.* denotes the special case of probability densities. For sake of clarity we show the most important relations only and do not claim completeness.

p corresponds to the observed data and q denotes the estimated or expected data. We assume $p(x)$ and $q(x)$ are positive measures defined on x in the domain V. The weight of the functional p is defined as

$$W(p) = \int_V p(x)\, dx \ . \tag{9.1}$$

Positive measures with the additional constraint $W(p) = 1$ can be interpreted as probability density functions. Generally speaking, divergences measure a quasi-distance or directed difference, while we are mostly interested in separable measures, which satisfy the condition

$$\mathrm{D}(p\|q) \begin{cases} > 0 \text{ for } p \neq q \\ = 0 \text{ iff } p \equiv q \ . \end{cases} \tag{9.2}$$

In contrast to a metric, divergences may be non-symmetric $\mathrm{D}(p\|q) \neq \mathrm{D}(q\|p)$, and do not necessarily satisfy the triangular inequality $\mathrm{D}(p\|q) \leqslant \mathrm{D}(p\|z) + \mathrm{D}(z\|q)$. Following (Cichocki et al. 2009) one can distinguish at least three main families of divergences with the same consistent properties: Bregman-divergences, Csiszár's f-divergences and Gamma-divergences. Note that all these families contain the KL divergence as special case, so the KL divergence can be seen as the non empty intersection between the sets of divergences.

In general we assume p and q to be positive measures. In case they are normalized we refer to them as probability densities. We review some basic properties of divergences in the following Sections. For detailed information see (Cichocki et al. 2009, Cichocki and Amari 2010). An overview of the family of divergences, examples and their relationship to each other can be found in Figure 9.1.

9.2.1 Bregman divergences

A Bregman divergence is defined as a pseudo-distance between two positive measures p and q: $D_{\mathrm{B}}(p\|q) : \mathcal{L} \times \mathcal{L} \rightarrow \mathbb{R}^{+}$. Let ϕ be a strictly convex real-valued function with the domain of the Lebesgue-integrable functions $\mathcal{L}$ and twice continuously Fréchet-differentiable (Kantorowitsch and Akilow 1978). Then the Bregman divergence can be defined by

$$D_{\mathrm{B}}^{\phi}(p\|q) = \phi(p) - \phi(q) - \frac{\delta\phi(q)}{\delta q}[p-q] \ , \tag{9.3}$$

where $\frac{\delta\phi(q)}{\delta q}$ is the Fréchet derivative of ϕ with respect to q (Villmann and Haase 2011). Well known fundamental properties of the Bregman divergences are (Cichocki et al. 2009):

Convexity A Bregman divergence is always convex in its first argument but not necessarily in its second.

Non-negativity

$$D_{\mathrm{B}}^{\phi}(p\|q) \geqslant 0 \text{ and } D_{\mathrm{B}}^{\phi}(p\|q) = 0 \text{ iff } p \equiv q \tag{9.4}$$

Linearity They are linear according to the generating function Φ, i.e. any positive linear combination of Bregman divergences is also a Bregman divergence:

$$\mathrm{D}_{\mathrm{B}}^{c_1\phi_1 + c_2\phi_2}(.) = c_1\mathrm{D}_{\mathrm{B}}^{\phi_1}(.) + c_2\mathrm{D}_{\mathrm{B}}^{\phi_2}(.) \qquad c_1, c_2 > 0$$

Invariance A Bregman divergence is invariant under affine transformations. Thus, $\mathrm{D}_{\mathrm{B}}^{\Gamma}(p\|q) = \mathrm{D}_{\mathrm{B}}^{\phi}(p\|q)$ is valid for any affine transformation

$$\Gamma(q) = \phi(q) + \Psi_g[q] + c \tag{9.5}$$

with linear operator

$$\Psi_g[q] = \frac{\delta\Gamma(g)}{\delta g} \cdot q - \frac{\delta\phi(g)}{\delta g} \cdot q \tag{9.6}$$

for positive measures g and q and scalar c.

Three-point property For any triple p, q, g of positive measures

$$D_{\mathrm{B}}^{\phi}(p\|g) = D_{\mathrm{B}}^{\phi}(p\|q) + D_{\mathrm{B}}^{\phi}(q\|g) + (p-q)\left(\frac{\delta\phi(q)}{\delta q} - \frac{\delta\phi(g)}{\delta g}\right) \text{ holds.}$$

Generalized Pythagorean theorem Let $P_\Omega(q) = \underset{\omega\in\Omega}{\arg\min}\, \mathrm{D}_{\mathrm{B}}^{\phi}(\omega\|q)$ be the Bregman projection onto the convex set Ω and $p \in \Omega$. The inequality:

$$\mathrm{D}_{\mathrm{B}}^{\phi}(p\|q) \geqslant \mathrm{D}_{\mathrm{B}}^{\phi}(p\|P_\Omega(q)) + \mathrm{D}_{\mathrm{B}}^{\phi}(P_\Omega(q)\|q) \quad \text{(generalized Pythagorean theorem)} \tag{9.7}$$

holds. If Ω is an affine set it holds with equality.

Optimality In (Banerjee et al. 2005) an optimality property is stated. Given a set S of positive measures p with mean $\mu = E[S]$ and $\mu \in S$ the unique minimizer $E_{p\in S}[\mathrm{D}(p\|q)]$ is minimum for $q = \mu$ if D is a Bregman divergence. This property favors the Bregman divergences for optimization and clustering problems (Banerjee et al. 2004, Bregman 1967, Dhillon and Sra 2005, Dhillon and Tropp 2007, Murata et al. 2004).

The Bregman divergence include many prominent dissimilarity measures like (Cichocki et al. 2009, Villmann and Haase 2011, Eguchi and Kano 2001):

- The generalized Kullback-Leibler (or I-) divergence for positive measures p and q:

$$\mathrm{D}_{\mathrm{GKL}}(p\|q) = \int p \log\left(\frac{p}{q}\right) dx - \int (p-q)\, dx \tag{9.8}$$

  using the generating function

$$\Phi(f) = \int (f \cdot \log f - f)\, dx \ . \tag{9.9}$$

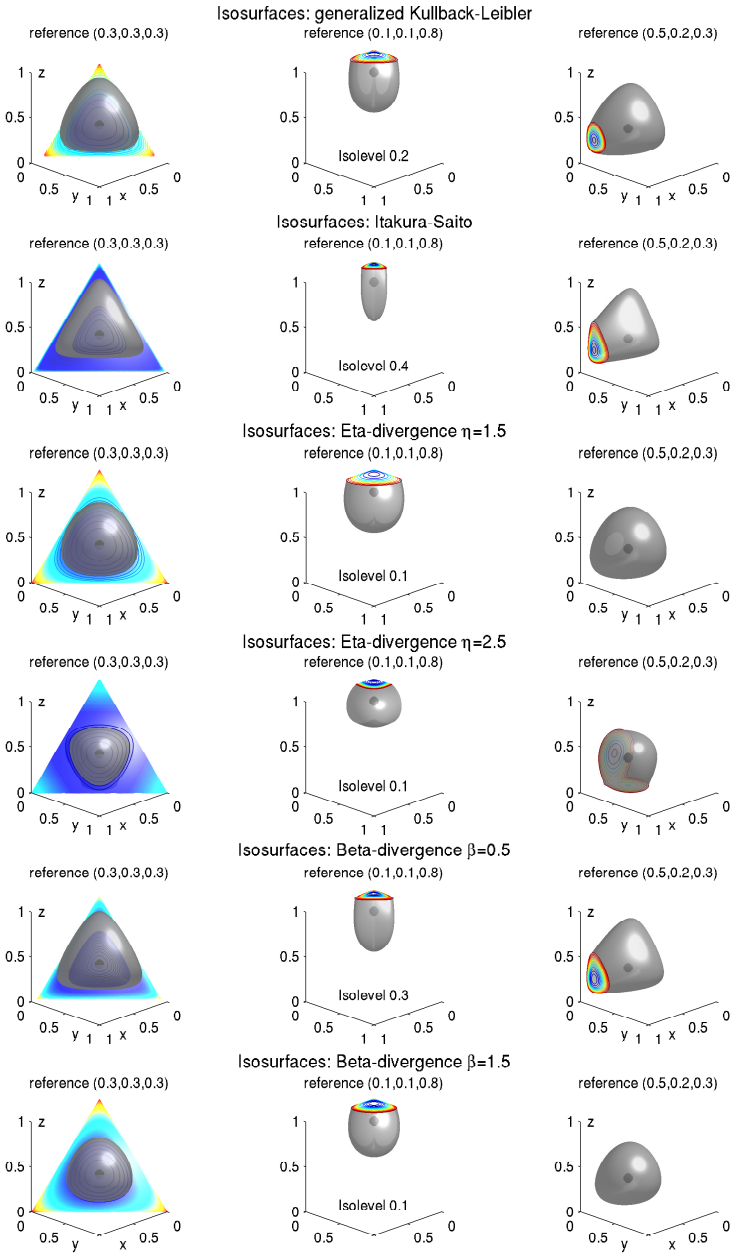


Figure 9.2: Isosurfaces of some Bregman divergences with respect to different reference points. The first panel of each row contains the plane of probability densities, the cutoffs in the other panels show the equidistance lines for this plane.

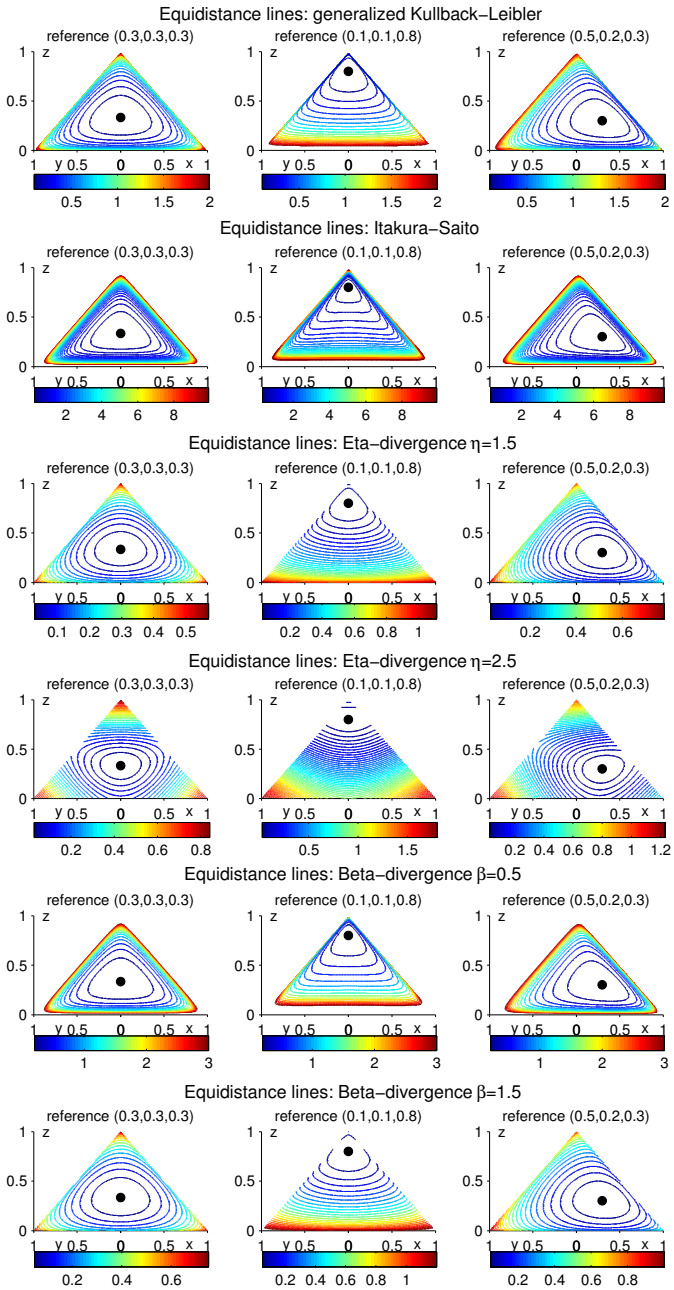


Figure 9.3: Equidistance lines of Bregman divergences for probability densities with respect to different reference points.

Some 3-dim. isosurfaces for the GKL divergence with respect to different reference points can be found in Figure 9.2. For probability densities p and q, Eq. (9.8) simplifies to the KL divergence (Kullback and Leibler 1951, Kapur 1994):

$$\mathrm{D}_{\mathrm{KL}}(p\|q) = \int p \, \log\left(\frac{p}{q}\right) \, dx \ , \tag{9.10}$$

which is related to the Shannon-entropy (Shannon 1948). Equidistance contours for 3-dim. probability densities using KL divergence with respect to different reference points are displayed in Fig. 9.3.

- The Itakura-Saito (IS) divergence (Itakura and Saito 1968) :

$$\mathrm{D}_{\mathrm{IS}}(p\|q) = \int \left[\frac{p}{q} - \log\left(\frac{p}{q}\right) - 1\right] dx \tag{9.11}$$

bases on the Burg entropy, which also serves as the generating function:

$$\Phi(f) = -\int \log(f) \, dx \ . \tag{9.12}$$

The IS divergence was originally presented as a measure of the quality of fits between two spectra and became a standard measure in the speech and image processing community due to the good perceptual properties of the reconstructed signals. It is known as negative cross-Burg entropy and fulfills the scale-invariance property $\mathrm{D}_{\mathrm{IS}}(c \cdot p\|c \cdot q) = \mathrm{D}_{\mathrm{IS}}(p\|q)$, which implies the same relative weight is given to low and high valued components of p (Bertin et al. 2009).

- The Eta-divergence is also known as norm-like divergence (Nielsen and Nock 2009) :

$$\mathrm{D}_{\eta}(p\|q) = \int p^{\eta} + (\eta - 1) \cdot q^{\eta} - \eta \cdot p \cdot q^{\eta - 1} \, dx \tag{9.13}$$

with generating function

$$\Phi(f) = \int f^{\eta} \, dx \text{ for } \eta > 1 \ . \tag{9.14}$$

In the case $\eta = 2$ the Eta-divergence becomes the Euclidean distance with generating function $\Phi(f) = \int f^2 \, dx$.

- The Beta-divergence (Cichocki et al. 2009):

$$\mathrm{D}_{\beta}(p\|q) = \int p \, \frac{p^{\beta-1} - q^{\beta-1}}{\beta - 1} \, dx - \int \frac{p^{\beta} - q^{\beta}}{\beta} \, dx \tag{9.15}$$

with $\beta \neq 0$ and $\beta \neq 1$ and the generating function

$$\Phi(f) = \frac{f^\beta - \beta \cdot f + \beta - 1}{\beta(\beta - 1)} \ . \tag{9.16}$$

For specific values of β the divergence becomes:
$\beta \to 1$: generalized Kullback-Leibler Eq. (9.8)
$\beta \to 0$: Itakura-Saito divergence Eq. (9.11)
$\beta = 2$: Euclidean distance (apart from a factor $\frac{1}{2}$).
Furthermore the Beta-divergence is equivalent to the density power divergence (Basu et al. 1998, Eguchi and Kano 2001, Mihoko and Eguchi 2002) and a rescaled version of the Eta-divergence.

9.2.2 Csiszár f-divergences

We denote by $\mathcal{F}$ the class of convex, real-valued, continuous functions f satisfying $f(1) = 0$, with

$$\mathcal{F} = \{g | g : [0, \infty) \to \mathbb{R}, g \text{ - convex}\} \ . \tag{9.17}$$

For a function $f \in \mathcal{F}$ the Csizár f-divergence is given by:

$$\mathrm{D}_f(p\|q) = \int q \cdot f\left(\frac{p}{q}\right) \, dx \tag{9.18}$$

with the definitions $0 \cdot f\left(\frac{0}{0}\right) = 0$ and $0 \cdot f\left(\frac{a}{0}\right) = \lim_{x \to 0} x \cdot f(\frac{a}{x}) = \lim_{u \to \infty} a \cdot \frac{f(u)}{u}$ (Csiszár 1967, Csiszár 1972, Amari and Nagaoka 2000, Taneja and Kumar 2004). The f-divergence can be interpreted as an average of the likelihood ratio $\frac{p}{q}$ describing the change rate of p with respect to q weighted by the determining function f. For a general f, which does not have to be convex, with $f'(1) = c_f \neq 0$, this form is not invariant and we need to use the generalized f-divergence

$$\mathrm{D}_f^{\mathrm{G}}(p\|q) = c_f \int (p - q) \, dx + \int q \, f\left(\frac{p}{q}\right) \, dx \ . \tag{9.19}$$

For the special case of probability densities p and q the first term vanishes and the original form of the f-divergences is obtained.

Some basic properties of the Csiszár f-divergence are (Österreicher 2002, Cichocki et al. 2009):

Non-negativity $\mathrm{D}_f(p\|q) \geqslant 0$ where the equal sign holds iff $p \equiv q$, which follows from the Jensen's inequality.

Generalized entropy It corresponds to a generalized f-entropy if the form

$$H_f(p) = -\int f(p(r))\, dr \ . \tag{9.20}$$

Strict convexity The f-divergence is convex in both arguments p and q:

$$\mathrm{D}_f(tp_1 + (1-t)p_2 \| tq_1 + (1-t)q_2) \leqslant \\ t\mathrm{D}_f(\tilde{p_1}\|\tilde{q_1}) + (1-t)\mathrm{D}_f(p_2\|q_2) \quad \forall t \in [0,1] \tag{9.21}$$

Scalability $c\mathrm{D}_f(p\|q) = \mathrm{D}_{cf}(p\|q)$ for any positive constant $c > 0$.

Invariance $\mathrm{D}_f(p\|q)$ is invariant with respect to a linear shift regarding the function f: e. g. $\mathrm{D}_f(p\|q) = \mathrm{D}_{\tilde{f}}(p\|q)$ iff $\tilde{f}(u) = f(u) + c \cdot (u-1)$ for any constant $c \in \mathbb{R}$.

Symmetry For $f, f^* \in \mathcal{F}$, where $f^*(u) = u \cdot f(\frac{1}{u})$ denotes the conjugate function of f, the relation $\mathrm{D}_f(p\|q) = \mathrm{D}_{f*}(q\|p)$ is valid. It is possible to construct a symmetric Csizár f-divergence with $f_{\mathrm{sym}}(u) = f(u) + f^*(u)$ as determining function.

Upper bound The f-divergence is bounded by

$$0 \leqslant \mathrm{D}_f(p\|q) \leqslant \lim_{u\to 0^+}\{f(u) + f^*(u)\} \text{ with } u = \frac{p}{q} \ . \tag{9.22}$$

The existence of this limit for probability densities p and q was shown by Liese and Vajda in (Liese and Vajda 1987). Villmann and Haase showed that these bounds still holds for positive measures p and q (Villmann and Haase 2011).

Monotonicity The f-divergence is monotonic with respect to the coarse-graining of the underlying domain $\mathcal{D}$ of the positive measures p and q, which is similar to the monotonicity of the Fisher metric (Amari and Nagaoka 2000).

Some well-known examples of f-divergences are (Cichocki et al. 2009):

- The subset of Alpha-divergences (Cichocki et al. 2009):

$$\mathrm{D}_\alpha(p\|q) = \frac{1}{\alpha(\alpha-1)} \cdot \int [p^\alpha q^{(1-\alpha)} - \alpha p + (\alpha-1)q]\, dx \tag{9.23}$$

  is based on the determining function

$$f(u) = u\frac{u^{(\alpha-1)}-1}{\alpha^2-\alpha} + \frac{1-u}{\alpha} \quad \text{with } u = \frac{p}{q} \tag{9.24}$$

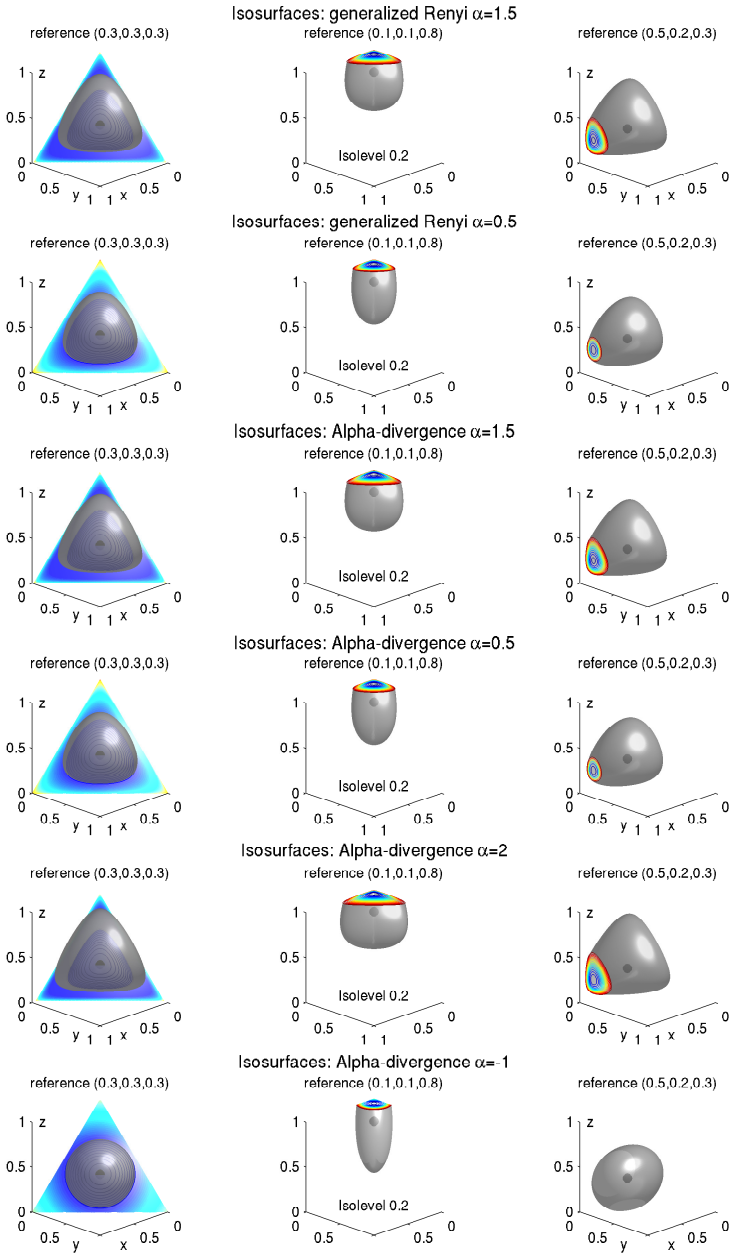


Figure 9.4: Isosurfaces of some Csiszár f-divergences with respect to different reference points. The first panel of each row contains the plane of probability densities, the cutoffs in the other panels show the equidistance lines for this plane.

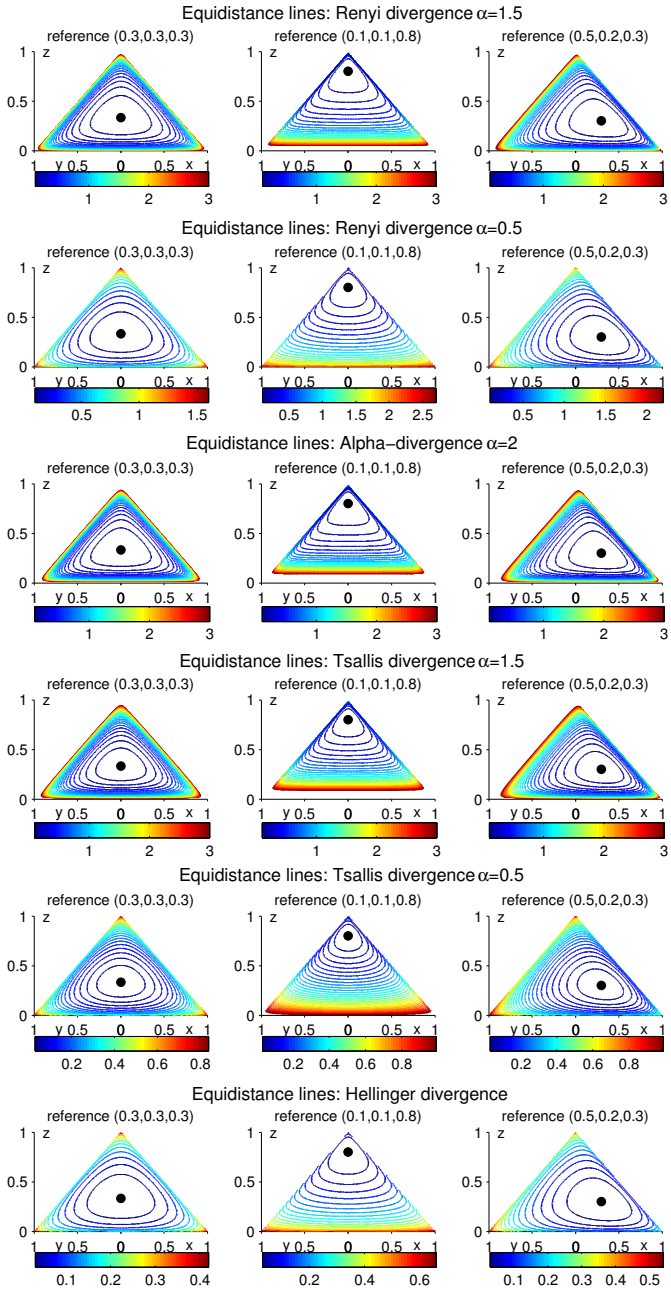


Figure 9.5: Equidistance lines of Csiszár f-divergences for probability densities with respect to different reference points.

with $\alpha \in \mathbb{R}\backslash\{0,1\}$. For specific values of α the divergence becomes:
$\alpha \to 1$: generalized Kullback-Leibler Eq. (9.8)
$\alpha \to 0$: reverse Kullback-Leibler
$\alpha = -1$: Neyman Chi-square
$\alpha = 2$: Pearson Chi-square.
For $\alpha \leqslant 0$ the divergence is zero-forcing, e.g. $p(x) = 0$ enforces $q(x) = 0$. On the other hand, for $\alpha \geqslant a$ it is zero-avoiding, i.e. $q(x) > 0$ whenever $p(x) > 0$. For $\alpha \to \infty$ $q(x)$ covers $p(x)$ completely and the Alpha-divergence is called inclusive in this case. Furthermore the Beta-divergences can be generated from the Alpha-divergences by applying a nonlinear transformation (Cichocki et al. 2009, Villmann and Haase 2011).

- The generalized Rényi divergence (Amari 1985, Cichocki et al. 2009):

$$\mathrm{D}^{\alpha}_{\mathrm{GR}}(p\|q) = \frac{1}{\alpha - 1} \cdot \log\left(\int \left[p^{\alpha} q^{(1-\alpha)} - \alpha p + (\alpha - 1)\, q\right]\, dx + 1\right) \tag{9.25}$$

  with $\alpha \in \mathbb{R}\backslash\{0,1\}$ is closely related to the Alpha-divergence.

- For the special case of probability densities the generalized Rényi-divergence reduces to the Rényi-divergence (Rényi 1960, Rényi 1970):

$$\mathrm{D}^{\alpha}_{\mathrm{R}}(p\|q) = \frac{1}{\alpha - 1} \cdot \log\left(\int p^{\alpha} q^{(1-\alpha)}\, dx\right) \tag{9.26}$$

  which bases on the Rényi entropy.

- The Tsallis-divergences

$$\mathrm{D}^{\alpha}_{\mathrm{T}}(p\|q) = \frac{1}{1 - \alpha}\left(1 - \int p^{\alpha} q^{(1-\alpha)}\, dx\right) \tag{9.27}$$

  for $\alpha \neq 1$ is a widely applied divergence for probability densities p and q based on the Tsallis entropy. It is also a rescaled version of the Alpha-divergence. In the limit $\alpha \to 1$ it converges to the Kullback-Leibler divergence Eq. (9.10).

- The Hellinger divergence (Taneja and Kumar 2004):

$$\mathrm{D}_{\mathrm{H}}(p\|q) = \frac{1}{2}\int (\sqrt{p} - \sqrt{q})^2\, dx \tag{9.28}$$

  with generating function $f(u) = 2(1 - \sqrt{u})$ for $u = \frac{p}{q}$ is defined for probability densities p and q.

9.2.3 Gamma-divergence

The Gamma-divergence is very robust with respect to outliers (Fujisawa and Eguchi 2008) and was proposed by Fujisawa and Eguchi:

$$\mathrm{D}_\gamma(p\|q) = \log\left[\frac{\left[\int p^{\gamma+1}\,dx\right]^{\frac{1}{\gamma^2+\gamma}} \cdot \left[\int q^{\gamma+1}\,dx\right]^{\frac{1}{\gamma+1}}}{\left(\int p\cdot q^{\gamma}\,dx\right)^{\frac{1}{\gamma}}}\right] \tag{9.29}$$

It is robust for $\gamma \in [0,1]$. In the limit $\gamma \to 0$ it becomes the Kullback-Leibler divergence $\mathrm{D}_{\mathrm{KL}}(p\|q)$ for probability densities. For $\gamma = 1$ it becomes the Cauchy-Schwarz divergence

$$\mathrm{D}_{\mathrm{CS}}(p\|q) = \frac{1}{2}\log\left(\int q^2\,dx \cdot \int p^2\,dx\right) - \log\left(\int p\cdot q\,dx\right)\ , \tag{9.30}$$

which is based on the quadratic Rényi-entropy. The Cauchy-Schwarz divergence is symmetric and was introduced considering the Cauchy-Schwarz inequality for norms. It is frequently applied for Parzen window estimation, especially suitable for spectral clustering as well as related graph cut problems (Principe et al. 2000, Jenssen 2005, Jenssen et al. 2006, Villmann and Haase 2011).

Some isosurfaces of the Gamma-divergence for different values of γ are shown in Fig. 9.6. The equidistance lines for the special case of probability densities can be found in Fig. 9.7. The Gamma-divergence displays some nice properties (Cichocki et al. 2009, Villmann and Haase 2011):

Invariance $\mathrm{D}_\gamma(p\|q)$ is invariant under scalar multiplication with positive constants

$$\mathrm{D}_\gamma(p\|q) = \mathrm{D}_\gamma(c_1\cdot p\|c_2\cdot q) \quad \forall c_1, c_2 > 0\ . \tag{9.31}$$

In case of positive measures the equation $\mathrm{D}_\gamma(p\|q) = 0$ holds only if $p = c\cdot q$ with $c > 0$. For probability densities $c = 1$ is required.

Pythagorean relation As for Bregman divergences a modified Pythagorean relation between positive measures can be stated for special choices of p, q, ρ. Let p be a distortion of q defined as convex combination with a positive distortion measure $\phi(r)$

$$p_\varepsilon(r) = (1-\varepsilon)\cdot q(r) + \varepsilon\cdot\phi(r)\ . \tag{9.32}$$

A positive measure g is denoted as ϕ-consistent if $\nu_g = \left(\int \phi(r)g(r)^\alpha\,dr\right)^{\frac{1}{\alpha}}$ is sufficiently small for large $\alpha > 0$. If two positive measures q and ρ are ϕ-consistent

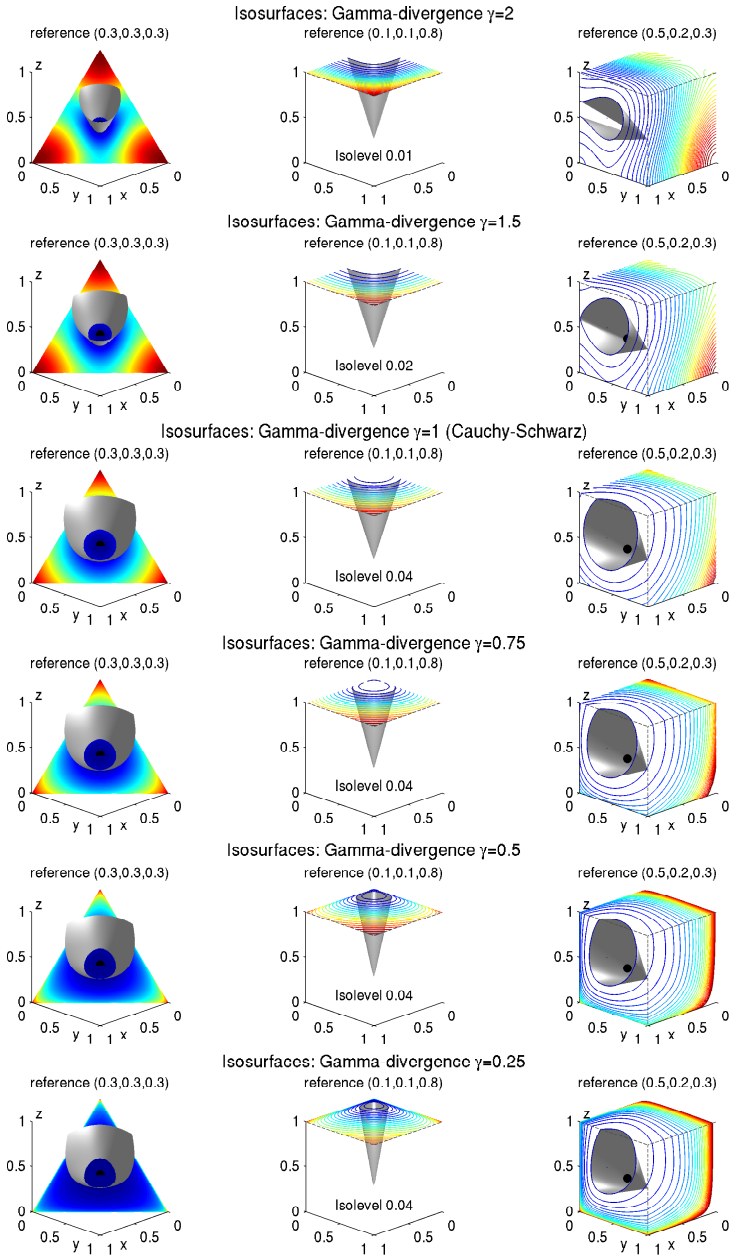


Figure 9.6: Isosurfaces of some Gamma-divergences with respect to different reference points. The first panel of each row contains the plane of probability densities, the other panels contain equidistance lines for certain limiting planes.

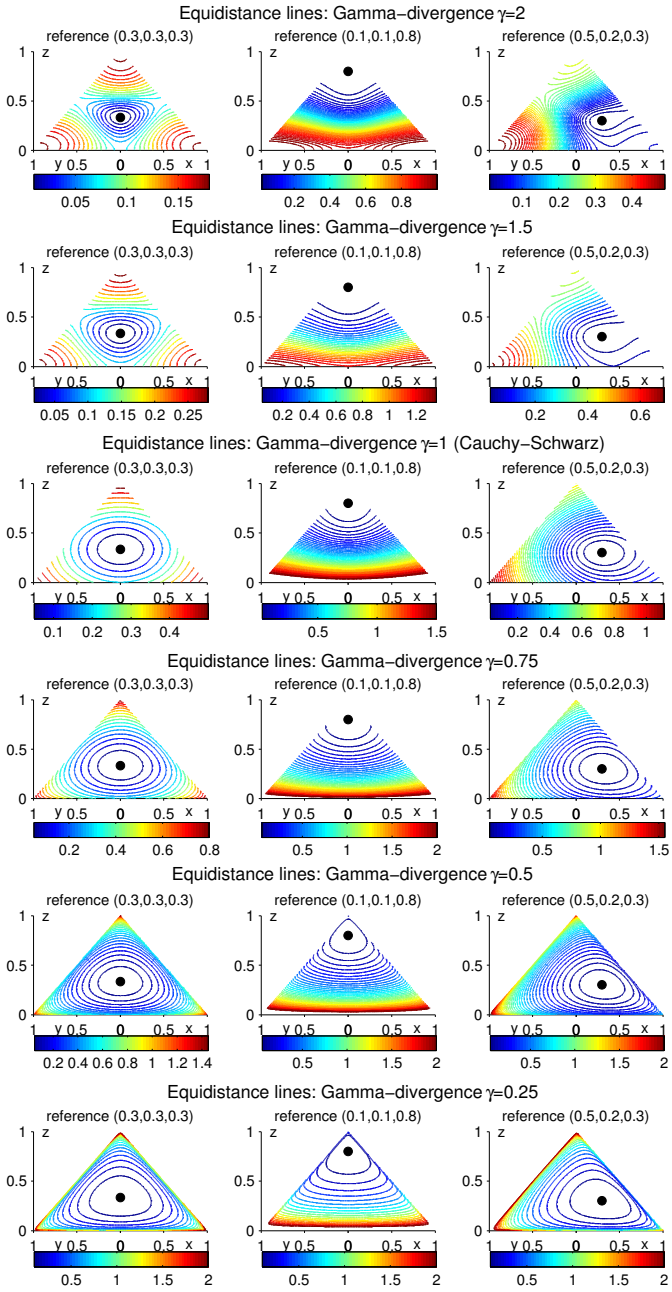


Figure 9.7: Equidistance lines of Gamma-divergences for probability densities with respect to different reference points.

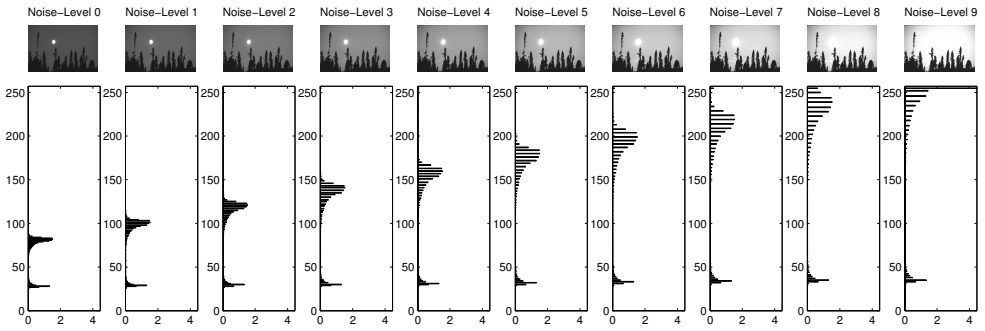


Figure 9.8: Histograms of intensity values in an example picture. The original image "moon" together with its histogram is shown on the left side. The following pictures contain noise in form of a linear monotonically increasing transformation of gray values (Eq. (9.34) using $l = [1, 2, \ldots, 9]$) corresponding to the Noise-Levels 1 till 9.

with respect to a distortion measure ϕ, then the Pythagorean relation approximately holds for q, ρ and the distortion p_ε of q:

$$\begin{aligned}\Delta(p_\varepsilon, q, \rho) &= \mathrm{D}_\gamma(p_\varepsilon \| \rho) - \mathrm{D}_\gamma(p_\varepsilon \| q) - \mathrm{D}_\gamma(q \| \rho) \\ &= \mathcal{O}(\varepsilon \nu^\gamma) \text{ with } \nu = \max\{\nu_q, \nu_\rho\}. \end{aligned} \tag{9.33}$$

This property implies the robustness of D_γ according to distortions.

9.2.4 Discussion of Divergences

In this section we examine and compare some introduced divergences by means of controlled experiments. We investigate the behavior of different divergences for the comparison of images containing an increasing level of (non-linear) noise. Therefore, we compute the histograms of gray-value images taken from the Berkley segmentation data set and noisy versions of them.

Linearly monotonically increasing noise

In the first experiment the noisy image I^* is obtained by adding a linear monotonically increasing transformation of gray values to the image I:

$$I^*(x, y) = I(x, y) \cdot [l \cdot (I(x, y) - I_0) + 1] \ , \tag{9.34}$$

where l denotes the level of noise and I_0 corresponds to the minimal intensity in the original image. Figure 9.8 shows the picture "moon" adding different levels

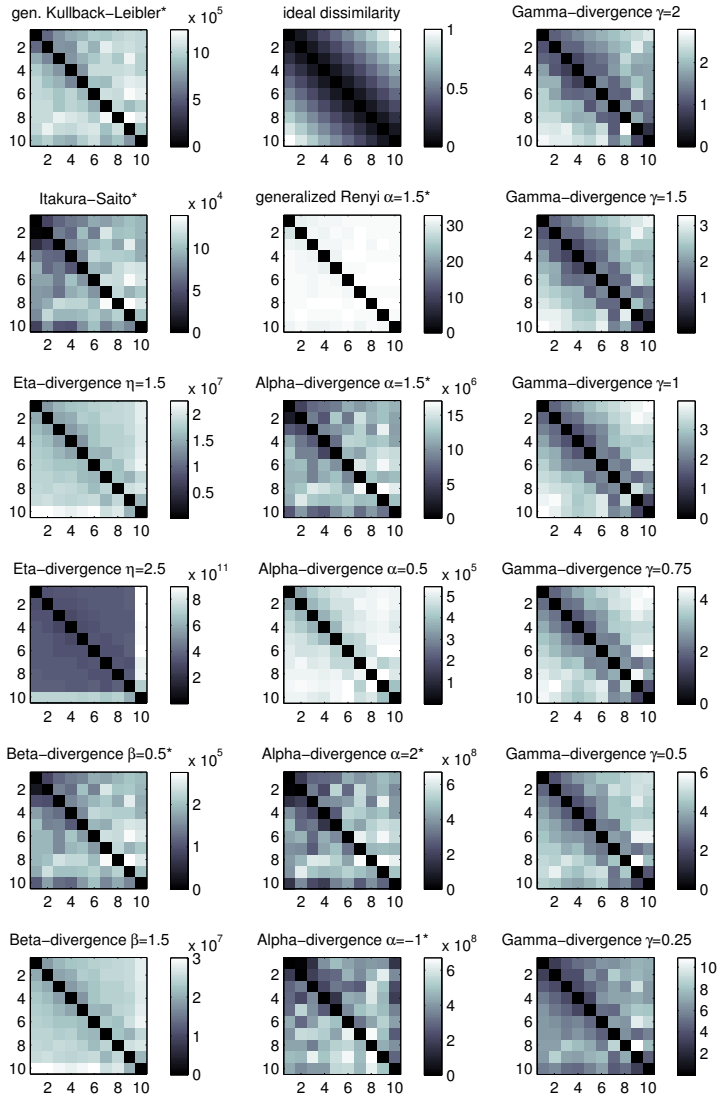


Figure 9.9: Matrix of pairwise dissimilarity of the ten histograms shown in figure 9.8 using different divergences. The ideal dissimilarity matrix for this example is a band matrix shown in the middle of the top row. Some divergences (marked with an asterisk * in the title) show numerical instabilities in case of zeros in the signals. In that cases a small constant $c = 1$ was added to all histograms to prevent the degeneration. Other divergences, like e.g. the Gamma-divergence are more robust. The Eta-divergence ignoring the extreme cases and the Gamma-divergence with $\gamma \geqslant 1$ exhibit more of the desired band structure for this example compared to other choices.

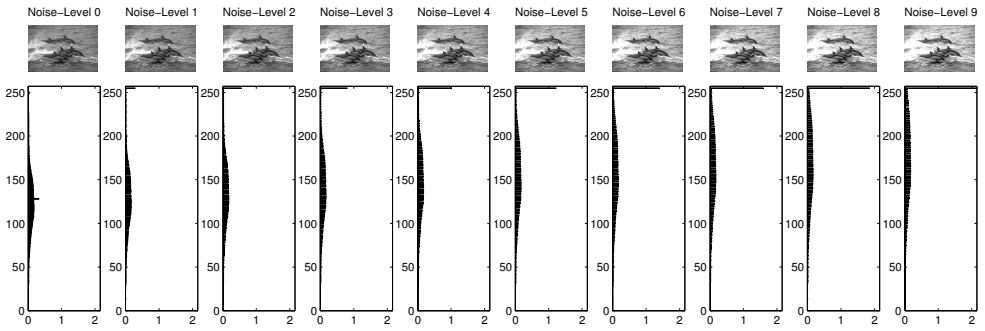


Figure 9.10: Histograms of intensity values in an example picture. The original image "dolphins" together with its histogram is shown on the left side. The following pictures contain noise in form of a linear monotonically increasing transformation of gray values (Eq. (9.34) using $l = [0.1, 0.2, \ldots, 0.9]$) named Noise-Levels 1 till 9.

of noise following Eq. (9.34) together with the gray-value histograms. The noise-level is ranged from $l = 1$ to $l = 9$. Some dissimilarity matrices comparing the ten histograms with different divergence measures are shown in Figure 9.9. The intuitively ideal dissimilarity matrix in this case is a symmetric band matrix shown in the middle of the top row. Some divergences like the generalized Rényi divergence show numerical instabilities. Others show quite similar behavior, e.g. Itakura Saito, Alpha-divergences and the Beta-divergence with $\beta = 0.5$, but they do not exhibit the desired band structure. For the original image and low noise-levels (images 1-5) the Beta-divergence with $\beta = 1.5$, Alpha-divergence with $\alpha = 0.5$ and also the generalized KL divergence show a bit of the desired band structure. Ignoring the last column and last row (the extreme case) in the dissimilarity matrix of the Eta-divergence shows a good approximation of a band matrix. The Gamma-divergence is observed to be quite robust in this case and also exhibits a visible band structure for $\gamma \geqslant 1$. In the special case of $\gamma = 1$ the Gamma-divergence equals the Cauchy-Schwarz divergence and is symmetric. Another symmetric example is the Alpha-divergence with $\alpha = 0.5$.

As a second example we take a picture of a group of dolphins and add some noise (following Eq. (9.34)) using the levels $l = [0.1, 0.2, \ldots, 0.9]$. The resulting histograms of gray values for the different noise levels are shown in Figure 9.10. As above we compute the matrices of pairwise similarities between the histograms using different divergences. The results can be found in Figure 9.11. In this example the eta-divergence especially with $\eta = 2.5$ is a good approximation of the ideal dissimilarity matrix shown in the middle of the top row. The best symmetric choice is the Gamma divergence with $\gamma = 1$ (Cauchy-Schwarz). Furthermore, dependent

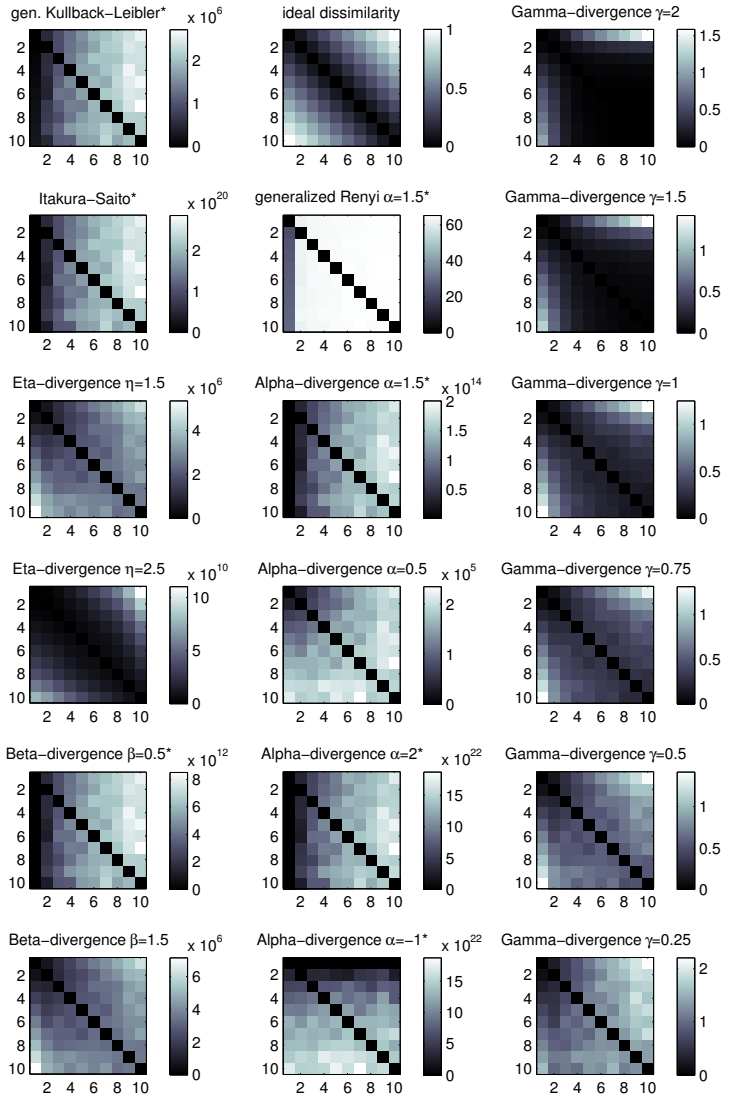


Figure 9.11: Matrix of pairwise dissimilarity of the ten histograms shown in figure 9.10 using different divergences. The ideal dissimilarity matrix for this example is a band matrix shown in the middle of the top row. Some divergences (marked with an asterisk * in the title) show numerical instabilities in case of zeros in the signals. In that cases a small constant $c = 1$ was added to all histograms to prevent the degeneration. The Eta-divergence especially with $\eta = 2.5$ shows a good approximation of the desired band structure for this example. The Gamma-divergence with $\gamma = 1$ (Cauchy-Schwarz) is the best symmetric choice in this case.

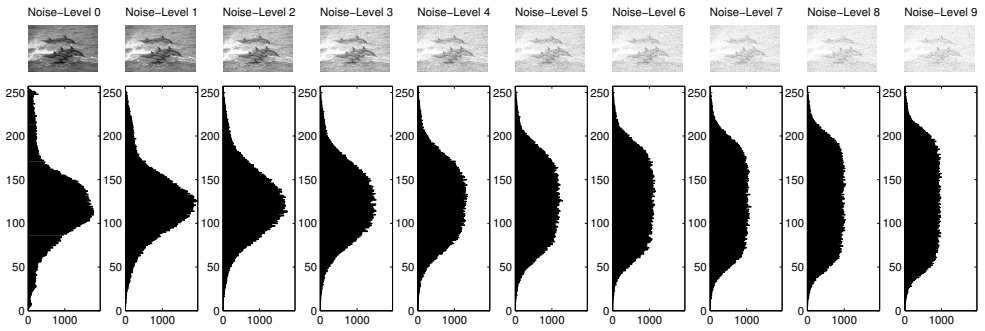


Figure 9.12: Histograms of intensity values in an example picture. The original image "dolphins" (top row) together with its histogram is shown on the left side. The following pictures contain additive uniform noise following Eq. (9.35) using $l = [\frac{50}{255}, \frac{100}{255}, \ldots, \frac{450}{255}]$ corresponding to the Noise-Levels 1 till 9.

on the value for γ one can chose between a better "resolution" (local) and a better preservation of the hierarchy of the histograms (global). Some other divergences, e.g. the generalized KL and Itakura-Saito, show very poor approximations of the desired dissimilarity for this example.

Additive uniform noise

In the second experiment the noisy image I^* is obtained by adding uniform noise to the image I:

$$I^*(x, y) = I(x, y) + \mathcal{U}(0, l) \ , \tag{9.35}$$

where $\mathcal{U}(0, l)$ denotes a scalar value drawn from the uniform distribution in the interval $[0, l]$.

Figure 9.12 shows the picture of dolphins adding different levels of uniform noise following Eq. (9.35) together with the more and more flattened gray-value histograms. The noise-level is ranged from $l = \frac{50}{255}$ to $l = \frac{450}{255}$. Some dissimilarity matrices pairwise comparing the ten images with different divergence measures are shown in Figure 9.13. Some divergences like the generalized Rényi, Itakura-Saito and some Alpha- and Beta-divergences fail to approximate the desired band structure in the pairwise dissimilarity matrix. Others, like the Gamma-, Eta- and some Alpha- and Beta-divergences are nearly ideal for this example. The Kullback-Leibler divergence is nearly perfect if the original image is ignored.

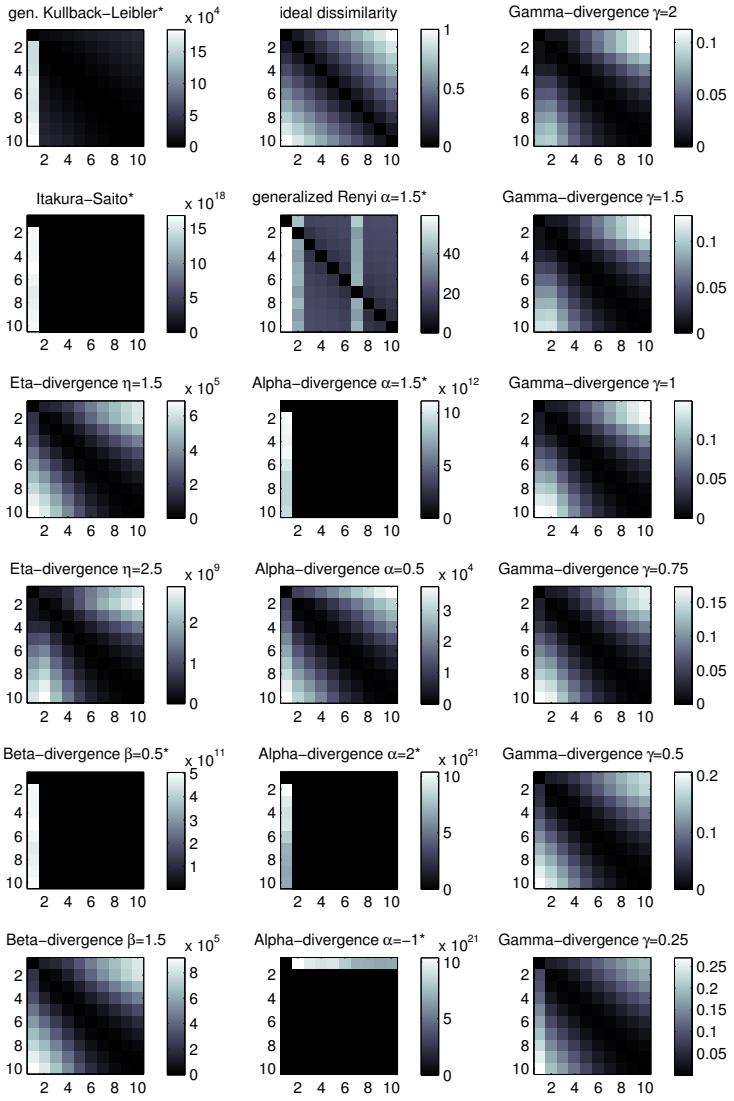


Figure 9.13: Dissimilarity matrices comparing the ten histograms shown in figure 9.12 using different divergences. The ideal dissimilarity matrix for this example is a band matrix shown in the middle of the top row. Some divergences (marked with an asterisk * in the title) show numerical instabilities in case of zeros in the signals. In that cases a small constant $c = 1$ was added to all histograms to prevent the degeneration. In this example the Eta-, Beta-, Gamma and the Alpha-divergences with $\alpha = 0.5$ show good approximations of the ideal band structure. Ignoring the original image also KL is nearly perfect. Other divergences like Itakura-Saito and generalized Rényi fail in this example.

9.3 The Fréchet Derivative

Suppose V and Z are Banach spaces and $U \subset V$ is an open subset of V. The function $f : U \to Z$ is called Fréchet differentiable at $x \in U$, if there exists a bounded linear operator $A_x : V \to Z$, such that for $h \in U$

$$\lim_{h \to 0} \frac{\|f(x+h) - f(x) - A_x(h)\|_Z}{\|h\|_V} = 0 \ . \tag{9.36}$$

This general definition can be used for functions $L : B \to \mathbb{R}$, defined as mappings from a functional Banach space B to $\mathbb{R}$. Further let B be equipped with a norm $\|\cdot\|$ and $f, h \in B$ are two functionals. The Fréchet derivative $\frac{\delta L[f]}{\delta f}$ of L at point f (i. e. in a function f) in the direction h is formally defined as:

$$\lim_{\epsilon \to 0} \frac{1}{\epsilon} \left(L[f + \epsilon h] - L[f] \right) =: \frac{\delta L[f]}{\delta f}[h] \ . \tag{9.37}$$

The Fréchet derivative in finite-dimensional spaces reduces to the usual partial derivative. Thus, it is a generalization of the directional derivatives.

Following (Villmann and Haase 2011) we introduce the functional derivatives of divergences in the next paragraphs. An overview is given in Table 9.1.

9.3.1 Fréchet derivatives of Bregman divergences

The Fréchet-derivative of $\mathrm{D}_\mathrm{B}^\phi$ Eq. (9.3) with respect to q is formally given by

$$\begin{aligned} &\frac{\delta \mathrm{D}_\mathrm{B}^\phi(p\|q)}{\delta q} = \frac{\delta \phi(p)}{\delta q} - \frac{\delta \phi(q)}{\delta q} - \frac{\delta \left[\frac{\delta \phi(q)}{\delta q}(p-q) \right]}{\delta q} \\ &\text{with} \\ &\frac{\delta \left[\frac{\delta \phi(q)}{\delta q}(p-q) \right]}{\delta q} = \frac{\delta^2[\phi(q)]}{\delta q^2}(p-q) - \frac{\delta \phi(q)}{\delta q} \ . \end{aligned} \tag{9.38}$$

For the generalized Kullback-Leibler divergence Eq. (9.8) this simplifies to

$$\frac{\delta \mathrm{D}_\mathrm{GKL}(p\|q)}{\delta q} = -\frac{p}{q} + 1 \ , \tag{9.39}$$

whereas for the Kullback-Leibler divergence Eq. (9.10) in the special case of probability densities it reads

$$\frac{\delta \mathrm{D}_\mathrm{KL}(p\|q)}{\delta q} = -\frac{p}{q} \ . \tag{9.40}$$

For the Itakura-Saito divergence Eq. (9.11) we get

$$\frac{\delta \mathrm{D}_{\mathrm{IS}}(p\|q)}{\delta q} = \frac{1}{q^2}(q-p) \tag{9.41}$$

and for the Eta-divergence Eq. (9.13) the Fréchet-derivative is

$$\frac{\delta \mathrm{D}_{\eta}(p\|q)}{\delta q} = q^{(\eta-2)} \cdot (1-\eta) \cdot \eta \cdot (p-q) \ . \tag{9.42}$$

In the case of $\eta = 2$ it reduces to the derivative of the Euclidean distance $-2(p-q)$. The Fréchet-derivative for the subset of Beta-divergences Eq. (9.15) is given by

$$\frac{\delta \mathrm{D}_{\beta}(p\|q)}{\delta q} = -p \cdot q^{(\beta-2)} + q^{(\beta-1)} = q^{(\beta-2)}(q-p) \ . \tag{9.43}$$

9.3.2 Fréchet derivatives of Csiszár f-divergences

For the Csiszár f-divergences Eq. (9.18) the Fréchet derivative is

$$\frac{\delta \mathrm{D}_{f}(p\|q)}{\delta q} = f\left(\frac{p}{q}\right) + q\frac{\partial f(u)}{\partial u}\frac{\delta u}{\delta q} = f\left(\frac{p}{q}\right) + q\frac{\partial f(u)}{\partial u} \cdot \frac{-p}{q^2} \ , \tag{9.44}$$

with $u = \frac{p}{q}$. For the set of Alpha-divergences Eq. (9.23) we get

$$\frac{\delta \mathrm{D}_{\alpha}(p\|q)}{\delta q} = -\frac{1}{\alpha}(p^{\alpha}q^{(-\alpha)} - 1) \ . \tag{9.45}$$

The related generalized Rényi divergence Eq. (9.25) yields

$$\frac{\delta \mathrm{D}^{\alpha}_{\mathrm{GR}}(p\|q)}{\delta q} = \frac{-p^{\alpha}q^{(-\alpha)} - 1}{\int [p^{\alpha}q^{(1-\alpha)} - \alpha p + (\alpha-1)q]dx + 1} \ , \tag{9.46}$$

which reduces in the case of the Rényi divergence for probability densities to

$$\frac{\delta \mathrm{D}^{\alpha}_{\mathrm{R}}(p\|q)}{\delta q} = \frac{-p^{\alpha}q^{(-\alpha)}}{\int p^{\alpha}q^{(1-\alpha)}\, dx} \ . \tag{9.47}$$

For the Tsallis divergence Eq. (9.27) the Fréchet derivative reads

$$\frac{\delta \mathrm{D}^{\alpha}_{\mathrm{T}}(p\|q)}{\delta q} = \frac{-p^{\alpha}q^{(-\alpha)}}{\int p^{\alpha}q^{(1-\alpha)}\, dx} \tag{9.48}$$

and for the well-known Hellinger divergence Eq. (9.28) the derivative is

$$\frac{\delta \mathrm{D}_{\mathrm{H}}(p\|q)}{\delta q} = 1 - \sqrt{\frac{p}{q}} \ . \tag{9.49}$$

Table 9.1: Table of divergences and their Fréchet derivatives.

Divergence family	Formula	Fréchet Derivative
Bregman	$\mathrm{D}_{\mathrm{B}}^{\phi}(p\|q) = \phi(p) - \phi(q) - \frac{\delta\phi(q)}{\delta q}[p-q]$	$\frac{\delta \mathrm{D}_{\mathrm{B}}^{\phi}(p\|q)}{\delta q} = \frac{\delta\phi(p)}{\delta q} - \frac{\delta\phi(q)}{\delta q} - \frac{\delta\left[\frac{\delta\phi(q)}{\delta q}(p-q)\right]}{\delta q}$
GKL	$\mathrm{D}_{\mathrm{GKL}}(p\|q) = \int p \cdot \log\left(\frac{p}{q}\right) dx - \int (p-q)\, dx$	$\frac{\delta \mathrm{D}_{\mathrm{GKL}}(p\|q)}{\delta q} = -\frac{p}{q} + 1$
Kullback-Leibler	$\mathrm{D}_{\mathrm{KL}}(p\|q) = \int p \cdot \log\left(\frac{p}{q}\right) dx$	$\frac{\delta \mathrm{D}_{\mathrm{KL}}(p\|q)}{\delta q} = -\frac{p}{q}$
Itakura-Saito	$\mathrm{D}_{\mathrm{IS}}(p\|q) = \int \left[\frac{p}{q} - \log\left(\frac{p}{q}\right) - 1\right] dx$	$\frac{\delta \mathrm{D}_{\mathrm{IS}}(p\|q)}{\delta q} = \frac{1}{q^2}(q-p)$
Eta-divergence	$\mathrm{D}_{\eta}(p\|q) = \int p^{\eta} + (\eta-1)\cdot q^{\eta} - \eta \cdot p \cdot q^{(\eta-1)}\, dx$	$\frac{\delta \mathrm{D}_{\eta}(p\|q)}{\delta q} = q^{(\eta-2)} \cdot (1-\eta) \cdot \eta \cdot (p-q)$
Beta-divergence	$\mathrm{D}_{\beta}(p\|q) = \int p\, \frac{p^{(\beta-1)} - q^{(\beta-1)}}{\beta-1}\, dx - \int \frac{p^{\beta} - q^{\beta}}{\beta}\, dx$	$\frac{\delta \mathrm{D}_{\beta}(p\|q)}{\delta q} = q^{(\beta-2)}(q-p)$
gen. Csiszár f	$\mathrm{D}_f^G(p\|q) = c_f \int (p-q)dx + \int q\, f\left(\frac{p}{q}\right) dx,\, c_f = f'(1) \neq 0$	$\frac{\delta \mathrm{D}_f^G(p\|q)}{\delta q} = f\left(\frac{p}{q}\right) q \frac{\partial f(u)}{\partial u} \cdot \frac{-p}{q^2},\, c_f = f'(1) \neq 0$
Csiszár f	$\mathrm{D}_f(p\|q) = \int q \cdot f\left(\frac{p}{q}\right) dx$	$\frac{\delta \mathrm{D}_f(p\|q)}{\delta q} = f\left(\frac{p}{q}\right) + q \frac{\partial f(u)}{\partial u} \cdot \frac{-p}{q^2}$
Alpha-divergence	$\mathrm{D}_{\alpha}(p\|q) = \frac{1}{\alpha(\alpha-1)} \cdot \int [p^{\alpha} q^{(1-\alpha)} - \alpha p + (\alpha-1) q]\, dx$	$\frac{\delta \mathrm{D}_{\alpha}(p\|q)}{\delta q} = -\frac{1}{\alpha}(p^{\alpha} q^{(-\alpha)} - 1)$
gen. Rényi	$\mathrm{D}_{\mathrm{GR}}^{\alpha}(p\|q) = \frac{1}{\alpha-1} \cdot \log\left(\int \left[p^{\alpha} q^{(1-\alpha)} - \alpha p + (\alpha-1)\, q\right] dx + 1\right)$	$\frac{\delta \mathrm{D}_{\mathrm{GR}}^{\alpha}(p\|q)}{\delta q} = \frac{-p^{\alpha} q^{(-\alpha)} - 1}{\int [p^{\alpha} q^{(1-\alpha)} - \alpha \cdot p + (\alpha-1) q] dx + 1}$
Rényi	$\mathrm{D}_{\mathrm{R}}^{\alpha}(p\|q) = \frac{1}{\alpha-1} \cdot \log\left(\int p^{\alpha} q^{(1-\alpha)}\, dx\right)$	$\frac{\delta \mathrm{D}_{\mathrm{R}}^{\alpha}(p\|q)}{\delta q} = \frac{-p^{\alpha} q^{(-\alpha)}}{\int p^{\alpha} q^{(1-\alpha)}\, dx}$
Tsallis	$\mathrm{D}_{\mathrm{T}}^{\alpha}(p\|q) = \frac{1}{1-\alpha}\left(1 - \int p^{\alpha} q^{(1-\alpha)}\, dx\right)$	$\frac{\delta \mathrm{D}_{\mathrm{T}}^{\alpha}(p\|q)}{\delta q} = \frac{-p^{\alpha} q^{(-\alpha)}}{\int p^{\alpha} q^{(1-\alpha)}\, dx}$
Hellinger	$\mathrm{D}_H(p\|q) = \frac{1}{2} \int (\sqrt{p} - \sqrt{q})^2\, dx$	$\frac{\delta \mathrm{D}_{\mathrm{H}}(p\|q)}{\delta q} = 1 - \sqrt{\frac{p}{q}}$
Gamma	$\mathrm{D}_{\gamma}(p\|q) = \log\left[\frac{\left(\int p^{(\gamma+1)}\, dx\right)^{\frac{1}{\gamma(\gamma+1)}} \cdot \left(\int q^{(\gamma+1)}\, dx\right)^{\frac{1}{\gamma+1}}}{\left(\int p \cdot q^{\gamma}\, dx\right)^{\frac{1}{\gamma}}}\right]$	$\frac{\delta \mathrm{D}_{\gamma}(p\|q)}{\delta q} = \frac{q^{\gamma}}{\int q^{(\gamma+1)}\, dx} - \frac{p \cdot q^{(\gamma-1)}}{\int p \cdot q^{\gamma}\, dx}$
Cauchy-Schwarz	$\mathrm{D}_{\mathrm{CS}}(p\|q) = \frac{1}{2} \cdot \log\left(\int q^2\, dx \cdot \int p^2\, dx\right) - \log\left(\int p \cdot q\, dx\right)$	$\frac{\delta \mathrm{D}_{\mathrm{CS}}(p\|q)}{\delta q} = \frac{q}{\int q^2\, dx} - \frac{p}{\int p \cdot q\, dx}$

9.3.3 Fréchet derivative of the Gamma-Divergence

The Fréchet derivative of the Gamma-divergence Eq. (9.29) can be written as

$$\frac{\delta \mathrm{D}_\gamma(p\|q)}{\delta q} = \frac{q^\gamma}{\int q^{(\gamma+1)}\, dx} - \frac{p \cdot q^{(\gamma-1)}}{\int p \cdot q^\gamma\, dx} \quad . \tag{9.50}$$

Considering the important special case $\gamma = 1$ the Cauchy-Schwarz divergence Eq. (9.30), the Fréchet derivative reads

$$\frac{\delta \mathrm{D}_{\mathrm{CS}}(p\|q)}{\delta q} = \frac{q}{\int q^2\, dx} - \frac{p}{\int p \cdot q\, dx} \quad . \tag{9.51}$$

9.4 Derivation of the general cost function gradient for t-SNE and SNE

Generally, dimensionality reduction methods convert a high dimensional data set $\mathbf{X} = \{\boldsymbol{x}, \boldsymbol{z}\} \in \mathbb{R}^N$ into low dimensional data $\boldsymbol{\Xi} = \{\boldsymbol{\xi}, \boldsymbol{\zeta}\} \in \mathbb{R}^M$. A probabilistic approach to visualize the structure of complex data sets, preserving neighbor similarities is SNE, proposed by (Hinton and Roweis 2003). In (van der Maaten and Hinton 2008) van der Maaten and Hinton presented a technique called t-SNE, which is a variation of SNE considering another statistical model assumption for data distributions. Both methods have in common that a probability distribution over all potential neighbors of a data point in the high-dimensional space is analyzed and described by their pairwise similarities. Both, t-SNE and the symmetric variant of SNE (van der Maaten and Hinton 2008) originally minimize the Kullback-Leibler divergence between a joint probability distribution in the high-dimensional space and its counterpart in the low-dimensional space as the underlying cost function, using a gradient descent method (see Algorithm 6.4 and Algorithm 6.5). We rewrite the pairwise similarities Eqs. (6.4) and (6.9) in the high-dimensional original data space:

$$p = p_{\boldsymbol{x}\boldsymbol{z}} = \frac{p_{\boldsymbol{z}|\boldsymbol{x}} + p_{\boldsymbol{x}|\boldsymbol{z}}}{2 \cdot \int 1\, d\boldsymbol{z}'} \tag{9.52}$$

with conditional probabilities

$$p_{\boldsymbol{z}|\boldsymbol{x}} = \frac{\exp\left(-\left\|\boldsymbol{x} - \boldsymbol{z}\right\|^2 / 2\sigma_{\boldsymbol{x}}^2\right)}{\int \exp\left(-\left\|\boldsymbol{x} - \boldsymbol{z}'\right\|^2 / 2\sigma_{\boldsymbol{x}}^2\right) d\boldsymbol{z}'} \, .$$

The variances $\sigma_{\boldsymbol{x}}$, which define the neighborhood corporation, are found by a line-search procedure parameterized by the so called perplexity. The perplexity is usually set to a value between 5 and 50 dependent on the data set size. Higher values

mean more neighbors are taken into account. For more information we refer to (van der Maaten and Hinton 2008). SNE and t-SNE differ in the model assumptions according to the distribution in the low-dimensional mapping space, defined more precisely in section 9.4.1.

9.4.1 The t-SNE gradient

Let $\mathrm{D}\left(p\|q\right)$ be a divergence for non-negative integrable measure functions $p = p\left(r\right)$ and $q = q(r)$ with a domain V and $\boldsymbol{\xi}, \boldsymbol{\zeta} \in \mathcal{E}$ distributed according to $\Pi_{\mathcal{E}}$ (Cichocki et al. 2009). Further, let $r\left(\boldsymbol{\xi}, \boldsymbol{\zeta}\right) : \mathcal{E} \times \mathcal{E} \rightarrow \mathbb{R}$ with the distribution $\Pi_r = \phi\left(r, \Pi_{\mathcal{E}}\right)$. We use the squared Euclidean distance in the low-dimensional space:

$$r = r_{\boldsymbol{\xi}\boldsymbol{\zeta}} = r\left(\boldsymbol{\xi}, \boldsymbol{\zeta}\right) = \left\|\boldsymbol{\xi} - \boldsymbol{\zeta}\right\|^2 \quad . \tag{9.53}$$

For t-SNE, q is obtained by means of a Student t-distribution, such that

$$q(r(\boldsymbol{\xi}', \boldsymbol{\zeta}')) = \frac{\left(1 + r\left(\boldsymbol{\xi}', \boldsymbol{\zeta}'\right)\right)^{-1}}{\int\int \left(1 + r\left(\boldsymbol{\xi}, \boldsymbol{\zeta}\right)\right)^{-1} d\boldsymbol{\xi} d\boldsymbol{\zeta}} \quad , \tag{9.54}$$

which we will abbreviate for reasons of clarity as

$$q(r') = \frac{\left(1 + r'\right)^{-1}}{\int\int \left(1 + r\right)^{-1} d\boldsymbol{\xi} d\boldsymbol{\zeta}} \quad . \tag{9.55}$$

The general t-SNE gradient is derived in Appendix 9.A and reads:

$$\begin{aligned} \frac{\partial \mathrm{D}}{\partial \boldsymbol{\xi}} =& 4 \int \frac{\delta \mathrm{D}}{\delta r} \left(\boldsymbol{\xi} - \boldsymbol{\zeta}\right) \, d\boldsymbol{\zeta} \\ =& 4 \int \frac{-q(r)}{1 + r} \left[\frac{\delta \mathrm{D}}{\delta q(r)} - \int \frac{\delta \mathrm{D}}{\delta q(r')} q(r') \Pi_{r'} \, dr'\right] \cdot \left(\boldsymbol{\xi} - \boldsymbol{\zeta}\right) \, d\boldsymbol{\zeta} \quad . \end{aligned} \tag{9.56}$$

We now have the obvious advantage that we can derive $\frac{\partial \mathrm{D}}{\partial \boldsymbol{\xi}}$ for several divergences $\mathrm{D}\left(p\|q\right)$ directly from Eq. (9.56), if the Fréchet derivative $\frac{\delta \mathrm{D}}{\delta q(r)}$ of D with respect to $q(r)$ is known.

9.4.2 The SNE gradient

In symmetric SNE, the pairwise similarities in the low dimensional-map are analogously defined following (van der Maaten and Hinton 2008)

$$q'_{\mathrm{SNE}} = q_{\mathrm{SNE}}\left(r\left(\boldsymbol{\xi}', \boldsymbol{\zeta}'\right)\right) = \frac{\exp\left(-r\left(\boldsymbol{\xi}', \boldsymbol{\zeta}'\right)\right)}{\int\int \exp\left(-r\left(\boldsymbol{\xi}, \boldsymbol{\zeta}\right)\right) d\boldsymbol{\xi} d\boldsymbol{\zeta}} \quad ,$$

which we will abbreviate below for reasons of clarity as

$$q_{\mathrm{SNE}}\left(r'\right) = \frac{\exp\left(-r'\right)}{\int\int \exp\left(-r\right) d\boldsymbol{\xi} d\boldsymbol{\zeta}} = g\left(r'\right) \cdot J^{-1} \,. \tag{9.57}$$

We obtain the general formulation of the SNE cost function gradient (Appendix 9.B):

$$\begin{aligned} \frac{\partial \mathrm{D}}{\partial \boldsymbol{\xi}} =& 4 \int \frac{\delta \mathrm{D}}{\delta r} \left(\boldsymbol{\xi} - \boldsymbol{\zeta}\right) \, d\boldsymbol{\zeta} \\ =& -4 \int q_{\mathrm{SNE}}\left(r\right) \left(\boldsymbol{\xi} - \boldsymbol{\zeta}\right) \cdot \left[\frac{\delta \mathrm{D}}{\delta q_{\mathrm{SNE}}\left(r\right)} - \int \frac{\delta \mathrm{D}}{\delta q_{\mathrm{SNE}}\left(r'\right)} q_{\mathrm{SNE}}\left(r'\right) \Pi_{r'} \, dr' \right] d\boldsymbol{\zeta} \end{aligned} \tag{9.58}$$

using the Fréchet-derivatives of the applied divergences as above for t-SNE.

9.5 t-SNE gradients for various divergences

In this section we explain the t-SNE gradients for various divergences. There exists a large variety of divergences, as mentioned in Section 9.2, which can be collected into several classes according to their mathematical properties and structural behavior. We extend the methods to arbitrary divergences by plug the corresponding Fréchet-derivatives into the general gradient Eq. (9.56) for t-SNE. Clearly, one can convey these results easily to the general SNE gradient Eq. (9.58) in complete analogy, because of its structural similarity to the t-SNE formula Eq. (9.56).

A technical remark should be made here: In the following we will abbreviate $p\left(r\right)$ by p and $p\left(r'\right)$ by p'. Further, because the integration variable r is a function $r = r\left(\boldsymbol{\xi}, \boldsymbol{\zeta}\right)$ an integration requires the weighting according to the distribution Π_r. Thus, the integration has formally to be carried out according to the differential $d\,\Pi_r\left(r\right)$ (Stieltjes-integral). We abbreviate this by dr but keeping this fact in mind, i.e. by this convention, we will drop the distribution Π_r, if it is clear from the context.

9.5.1 Bregman divergences

In the following we will provide the gradients for some examples of Bregman divergences introduced in Section 9.2.1. As a first example we show that we obtain the same result as (van der Maaten and Hinton 2008) for the Kullback-Leibler divergence Eq. (9.10). The Fréchet-derivative of D_{KL} with respect to q is given in Eq. (9.40). From Eq. (9.56) we see that

$$\begin{aligned} \frac{\partial \mathrm{D}_{\mathrm{KL}}}{\partial \boldsymbol{\xi}} =& 4 \int \frac{q(\boldsymbol{\xi} - \boldsymbol{\zeta})}{(1+r)} \left(\frac{p}{q} - \int \frac{p'}{q'} q \prime \Pi_{r'} \, dr' \right) \, d\boldsymbol{\zeta} \\ =& 4 \int \frac{q(\boldsymbol{\xi} - \boldsymbol{\zeta})}{(1+r)} \left(\frac{p}{q} - \int p' \Pi_{r'} \, dr' \right) \, d\boldsymbol{\zeta}. \end{aligned} \tag{9.59}$$

Since the Integral $I = \int p' \Pi_{r'} \, dr'$ in Eq. (9.59) can be written as an double integral over all pairs of data points $I = \int\int p' d\boldsymbol{\xi}' d\boldsymbol{\zeta}'$, we see from Eq. (9.52) that the integral I equals 1. So, Eq. (9.59) simplifies to

$$\begin{aligned}\frac{\partial \mathrm{D_{KL}}}{\partial \boldsymbol{\xi}} =& 4 \int \frac{q}{(1+r)} \left(\frac{p}{q} - 1 \right) (\boldsymbol{\xi} - \boldsymbol{\zeta}) \; d\boldsymbol{\zeta} \\ =& 4 \int (1+r)^{-1} (p-q) (\boldsymbol{\xi} - \boldsymbol{\zeta}) \; d\boldsymbol{\zeta} \; . \end{aligned} \tag{9.60}$$

This is exactly the differential form of the discrete version as proposed for t-SNE in (van der Maaten and Hinton 2008).

The Kullback-Leibler divergence used in original SNE and t-SNE belongs to the more general class of Bregman divergences (Bregman 1967). Another representative of this class of divergences is the Itakura-Saito divergence D_{IS} Eq. (9.11) with the Fréchet-derivative Eq. (9.41). For the calculation of the gradient $\frac{\partial \mathrm{D_{IS}}}{\partial \boldsymbol{\xi}}$ we substitute the Fréchet-derivative in Eq. (9.56) and obtain

$$\frac{\partial \mathrm{D_{IS}}}{\partial \boldsymbol{\xi}} = -4 \int \frac{q}{1+r} \left(\frac{1}{q^2} (q-p) - \int \frac{q'-p'}{q'} \Pi_{r'} \, dr' \right) (\boldsymbol{\xi} - \boldsymbol{\zeta}) \; d\boldsymbol{\zeta} \tag{9.61}$$

$$= \int \frac{4 (\boldsymbol{\xi} - \boldsymbol{\zeta})}{1+r} \left(\frac{p}{q} - 1 + q \int \left[1 - \frac{p'}{q'} \right] \Pi_{r'} \, dr' \right) d\boldsymbol{\zeta}. \tag{9.62}$$

One more Bregman divergence is the norm-like or Eta-divergence Eq. (9.13). The Fréchet-derivative of D_η with respect to q is given in Eq. (9.42). Again, we are interested in the gradient $\frac{\partial \mathrm{D}_\eta}{\partial \boldsymbol{\xi}}$, which is

$$\frac{\partial \mathrm{D}_\eta}{\partial \boldsymbol{\xi}} = 4\eta(\eta-1) \int \frac{\boldsymbol{\xi} - \boldsymbol{\zeta}}{1+r} \left((p-q) q^{\eta-1} - q \cdot \int \left(p' - q' \right) q'^{(\eta-1)} \Pi_{r'} \, dr' \right) \; d\boldsymbol{\zeta} \; . \tag{9.63}$$

The last example of Bregman divergences we handle in this paper is the class of Beta-divergences defined in Eq. (9.15). We use Eq. (9.56) and insert the Fréchet-derivative of the Beta-divergences, given by Eq. (9.43). Thereby the gradient $\frac{\partial \mathrm{D}_\beta}{\partial \boldsymbol{\xi}}$ reads as

$$\frac{\partial \mathrm{D}_\beta}{\partial \boldsymbol{\xi}} = 4 \int \frac{\boldsymbol{\xi} - \boldsymbol{\zeta}}{1+r} \left(q^{\beta-1} (p-q) - q \cdot \int q'^{(\beta-1)} \left(p' - q' \right) \; \Pi_{r'} \, dr' \right) d\boldsymbol{\zeta} \; . \tag{9.64}$$

9.5.2 Csiszár's f-divergences

Now we will consider some divergences belonging to the class of Csiszár's f-divergences (see Section 9.2.2). A well-known example is the Hellinger divergence defined in Eq.

(9.28), with the Fréchet-derivative Eq. (9.49). The gradient of $\mathrm{D_H}$ with respect to $\boldsymbol{\xi}$ is

$$\begin{aligned}\frac{\partial \mathrm{D_H}}{\partial \boldsymbol{\xi}} =& 4 \int \frac{1}{1+r} \left(\sqrt{p\, q} - q - q \int \left(\sqrt{p' q'} - q' \right) \Pi_{r'}\, dr' \right) (\boldsymbol{\xi} - \boldsymbol{\zeta})\, d\boldsymbol{\zeta} \\ =& 4 \int \frac{\boldsymbol{\xi} - \boldsymbol{\zeta}}{1+r} \left(\sqrt{p\, q} - q \int \sqrt{p' q'} \Pi_{r'}\, dr' \right) d\boldsymbol{\zeta}. \end{aligned} \tag{9.65}$$

For the Alpha-divergence, see Eqs. (9.23) and (9.45), we get

$$\begin{aligned}\frac{\partial \mathrm{D}_\alpha}{\partial \boldsymbol{\xi}} =& \frac{4}{\alpha} \int \frac{q(\boldsymbol{\xi} - \boldsymbol{\zeta})}{1+r} \left(p^\alpha q^{(-\alpha)} - 1 - \int \left(p'^\alpha q'^{(-\alpha)} - 1 \right) q' \Pi_{r'}\, dr' \right) d\boldsymbol{\zeta} \\ =& \frac{4}{\alpha} \int \frac{\boldsymbol{\xi} - \boldsymbol{\zeta}}{1+r} \left(p^\alpha q^{(1-\alpha)} - q \int p'^\alpha q'^{(1-\alpha)} \Pi_{r'}\, dr' \right) d\boldsymbol{\zeta}. \end{aligned} \tag{9.66}$$

For the Tsallis divergence, Eqs. (9.27) and (9.48), we get

$$\begin{aligned}\frac{\partial \mathrm{D}_\alpha^{\mathrm{T}}}{\partial \boldsymbol{\xi}} =& \int \frac{4(\boldsymbol{\xi} - \boldsymbol{\zeta}) q}{1+r} \left(\left[\frac{p}{q} \right]^\alpha - \int \left[\frac{p'}{q'} \right]^\alpha q' \Pi_{r'} dr' \right) d\boldsymbol{\zeta} \\ =& 4 \int \frac{\boldsymbol{\xi} - \boldsymbol{\zeta}}{1+r} \left(p^\alpha q^{(1-\alpha)} - q \int p'^\alpha q'^{(1-\alpha)} \Pi_{r'} dr' \right) d\boldsymbol{\zeta}, \end{aligned} \tag{9.67}$$

which is also clear from Eq. (9.66), since the Tsallis divergence is a rescaled version of the Alpha-divergence for probability densities.
For the Rényi divergence, Eqs. (9.26) and (9.47), the derivative reads

$$\begin{aligned}\frac{\partial \mathrm{D}_\mathrm{R}^\alpha}{\partial \boldsymbol{\xi}} =& \frac{4}{\int p'^\alpha q'^{(1-\alpha)} dr'} \int \frac{\boldsymbol{\xi} - \boldsymbol{\zeta}}{1+r} \cdot \left(p^\alpha q^{1-\alpha} - q \int p'^\alpha q'^{(1-\alpha)} \Pi_{r'}\, dr' \right) d\boldsymbol{\zeta} \\ =& 4 \int \frac{\boldsymbol{\xi} - \boldsymbol{\zeta}}{1+r} \left(\frac{p^\alpha q^{(1-\alpha)}}{\int p'^\alpha q'^{(1-\alpha)} dr'} - q \right) d\boldsymbol{\zeta}\ . \end{aligned} \tag{9.68}$$

9.5.3 Gamma-divergence

The Fréchet-derivative of $\mathrm{D}_\gamma\,(p \| q)$ with respect to q is given in Eq. (9.29) and can be rewritten as

$$\begin{aligned}\frac{\delta \mathrm{D}_\gamma\,(p\|q)}{\delta q} =& q^{(\gamma-1)} \left[\frac{q}{\int q^{(\gamma+1)} dr} - \frac{p}{\int p\, q^\gamma dr} \right] = \frac{q^\gamma}{Q_\gamma} - \frac{p\, q^{(\gamma-1)}}{V_\gamma} \\ =& \frac{q^\gamma V_\gamma - p\, q^{(\gamma-1)} Q_\gamma}{Q_\gamma V_\gamma}\ . \end{aligned}$$

Table 9.2: Table of divergences and their t-SNE gradients.

Divergence family	Functional gradient for t-SNE	Gradients for discrete data $\{\boldsymbol{x}\}_{i=1}^n \in \mathbb{R}^N$ and $\{\boldsymbol{\xi}\}_{i=1}^n \in \mathbb{R}^M$
$\mathrm{D}_{\mathrm{KL}}(p\|q)$ Eq. (9.10)	$\frac{\partial \mathrm{D}_{\mathrm{KL}}}{\partial \boldsymbol{\xi}} = 4 \int \frac{\boldsymbol{\xi}-\boldsymbol{\zeta}}{1+r} \, (p-q) \, d\boldsymbol{\zeta}$	$\frac{\partial \mathrm{D}_{\mathrm{KL}}}{\partial \boldsymbol{\xi}^i} = 4 \sum_j^n \frac{\boldsymbol{\xi}^i-\boldsymbol{\xi}^j}{1+r_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}} \left(p_{\boldsymbol{x}^i\boldsymbol{x}^j} - q_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}\right)$
$\mathrm{D}_{\mathrm{IS}}(p\|q)$ Eq. (9.11)	$\frac{\partial \mathrm{D}_{\mathrm{IS}}}{\partial \boldsymbol{\xi}} = 4 \int \frac{\boldsymbol{\xi}-\boldsymbol{\zeta}}{1+r} \left[\frac{p}{q} - 1 + q \int \left[1 - \frac{p'}{q'}\right] \Pi_{r'} \, dr'\right] d\boldsymbol{\zeta}$	$\frac{\partial \mathrm{D}_{\mathrm{IS}}}{\partial \boldsymbol{\xi}^i} = 4 \sum_j^n \frac{\boldsymbol{\xi}^i-\boldsymbol{\xi}^j}{1+r_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}} \left(\frac{p_{\boldsymbol{x}^i\boldsymbol{x}^j}}{q_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}} - 1 + q_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j} \sum_{kl} \left(1 - \frac{p_{\boldsymbol{x}^k\boldsymbol{x}^l}}{q_{\boldsymbol{\xi}^k\boldsymbol{\xi}^l}}\right)\right)$
$\mathrm{D}_{\eta}(p\|q)$ Eq. (9.13)	$\frac{\partial \mathrm{D}_{\eta}}{\partial \boldsymbol{\xi}} = 4(\eta^2-\eta) \int \frac{\boldsymbol{\xi}-\boldsymbol{\zeta}}{1+r} \Bigg((p-q) q^{(\eta-1)} - q \cdot \int \left[p' - q'\right] q'^{(\eta-1)} \Pi_{r'} \, dr' \Bigg) \, d\boldsymbol{\zeta}$	$\frac{\partial \mathrm{D}_{\eta}}{\partial \boldsymbol{\xi}^i} = 4(\eta^2-\eta) \sum_j^n \frac{\boldsymbol{\xi}^i-\boldsymbol{\xi}^j}{1+r_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}} \Bigg(\left(p_{\boldsymbol{x}^i\boldsymbol{x}^j} - q_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}\right) q_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}^{(\eta-1)} - q_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j} \cdot \sum_{kl} \left(p_{\boldsymbol{x}^k\boldsymbol{x}^l} - q_{\boldsymbol{\xi}^k\boldsymbol{\xi}^l}\right) q_{\boldsymbol{\xi}^k\boldsymbol{\xi}^l}^{(\eta-1)} \Bigg)$
$\mathrm{D}_{\beta}(p\|q)$ Eq. (9.15)	$\frac{\partial \mathrm{D}_{\beta}}{\partial \boldsymbol{\xi}} = 4 \int \frac{\boldsymbol{\xi}-\boldsymbol{\zeta}}{1+r} \Bigg[q^{\beta-1} (p-q) - q \int q'^{(\beta-1)} \left(p' - q'\right) \Pi_{r'} \, dr' \Bigg] d\boldsymbol{\zeta}$	$\frac{\partial \mathrm{D}_{\beta}}{\partial \boldsymbol{\xi}^i} = 4 \sum_j^n \frac{\boldsymbol{\xi}^i-\boldsymbol{\xi}^j}{1+r_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}} \Bigg(q_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}^{\beta-1} \left(p_{\boldsymbol{x}^i\boldsymbol{x}^j} - q_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}\right) - q_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j} \sum_{kl} q_{\boldsymbol{\xi}^k\boldsymbol{\xi}^l}^{(\beta-1)} \left(p_{\boldsymbol{x}^k\boldsymbol{x}^l} - q_{\boldsymbol{\xi}^k\boldsymbol{\xi}^l}\right) \Bigg)$
$\mathrm{D}_{\alpha}(p\|q)$ Eq. (9.23)	$\frac{\partial \mathrm{D}_{\alpha}}{\partial \boldsymbol{\xi}} = \frac{4}{\alpha} \int \frac{\boldsymbol{\xi}-\boldsymbol{\zeta}}{1+r} \left(p^{\alpha} q^{1-\alpha} - q \int p'^{\alpha} q'^{(1-\alpha)} \Pi_{r'} \, dr'\right) d\boldsymbol{\zeta}$	$\frac{\partial \mathrm{D}_{\alpha}}{\partial \boldsymbol{\xi}^i} = \frac{4}{\alpha} \sum_j^n \frac{\boldsymbol{\xi}^i-\boldsymbol{\xi}^j}{1+r_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}} \left(p_{\boldsymbol{x}^i\boldsymbol{x}^j}^{\alpha} q_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}^{1-\alpha} - q_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j} \sum_{kl} p_{\boldsymbol{x}^k\boldsymbol{x}^l}^{\alpha} q_{\boldsymbol{\xi}^k\boldsymbol{\xi}^l}^{(1-\alpha)}\right)$
$\mathrm{D}_{\alpha}^{\mathrm{R}}(p\|q)$ Eq. (9.26)	$\frac{\partial \mathrm{D}_{\alpha}^{\mathrm{R}}}{\partial \boldsymbol{\xi}} = 4 \int \frac{\boldsymbol{\xi}-\boldsymbol{\zeta}}{1+r} \left(\frac{p^{\alpha} q^{1-\alpha}}{\int p'^{\alpha} q'^{(1-\alpha)} dr'} - q\right) d\boldsymbol{\zeta}$	$\frac{\partial \mathrm{D}_{\alpha}^{\mathrm{R}}}{\partial \boldsymbol{\xi}^i} = 4 \sum_j^n \frac{\boldsymbol{\xi}^i-\boldsymbol{\xi}^j}{1+r_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}} \left(\frac{p_{\boldsymbol{x}^i\boldsymbol{x}^j}^{\alpha} q_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}^{1-\alpha}}{\sum_{kl} p_{\boldsymbol{x}^k\boldsymbol{x}^l}^{\alpha} q_{\boldsymbol{\xi}^k\boldsymbol{\xi}^l}^{(1-\alpha)}} - q_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}\right)$
$\mathrm{D}_{\alpha}^{\mathrm{T}}(p\|q)$ Eq. (9.27)	$\frac{\partial \mathrm{D}_{\alpha}^{\mathrm{T}}}{\partial \boldsymbol{\xi}} = 4 \int \frac{\boldsymbol{\xi}-\boldsymbol{\zeta}}{1+r} \left(p^{\alpha} q^{(1-\alpha)} - q \int p'^{\alpha} q'^{(1-\alpha)} \Pi_{r'} \, dr'\right) d\boldsymbol{\zeta}$	$\frac{\partial \mathrm{D}_{\alpha}^{\mathrm{T}}}{\partial \boldsymbol{\xi}^i} = 4 \sum_j^n \frac{\boldsymbol{\xi}^i-\boldsymbol{\xi}^j}{1+r_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}} \left(p_{\boldsymbol{x}^i\boldsymbol{x}^j}^{\alpha} q_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}^{(1-\alpha)} - q \sum_{kl} p_{\boldsymbol{x}^k\boldsymbol{x}^l}^{\alpha} q_{\boldsymbol{\xi}^k\boldsymbol{\xi}^l}^{(1-\alpha)}\right)$
$\mathrm{D}_{\mathrm{H}}(p\|q)$ Eq. (9.28)	$\frac{\partial \mathrm{D}_{\mathrm{H}}}{\partial \boldsymbol{\xi}} = 4 \int \frac{\boldsymbol{\xi}-\boldsymbol{\zeta}}{1+r} \left(\sqrt{p\,q} - q \int \sqrt{p' q'} \, \Pi_{r'} \, dr'\right) d\boldsymbol{\zeta}$	$\frac{\partial \mathrm{D}_{\mathrm{H}}}{\partial \boldsymbol{\xi}^i} = 4 \sum_j^n \frac{\boldsymbol{\xi}^i-\boldsymbol{\xi}^j}{1+r_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}} \left(\sqrt{p_{\boldsymbol{x}^i\boldsymbol{x}^j} q_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}} - q_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j} \sum_{kl} \sqrt{p_{\boldsymbol{x}^k\boldsymbol{x}^l} q_{\boldsymbol{\xi}^k\boldsymbol{\xi}^l}}\right)$
$\mathrm{D}_{\gamma}(p\|q)$ Eq. (9.29)	$\frac{\partial \mathrm{D}_{\gamma}}{\partial \boldsymbol{\xi}} = 4 \int \frac{\boldsymbol{\xi}-\boldsymbol{\zeta}}{1+r} \left(\frac{p\, q^{\gamma}}{\int p' q'^{\gamma} dr'} - \frac{q^{(\gamma+1)}}{\int q'^{(\gamma+1)} dr'}\right) d\boldsymbol{\zeta}$	$\frac{\partial \mathrm{D}_{\gamma}}{\partial \boldsymbol{\xi}^i} = 4 \sum_j^n \frac{\boldsymbol{\xi}^i-\boldsymbol{\xi}^j}{1+r_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}} \left(\frac{p_{\boldsymbol{x}^i\boldsymbol{x}^j}\, q_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}^{\gamma}}{\sum_{kl} p_{\boldsymbol{x}^k\boldsymbol{x}^l} q_{\boldsymbol{\xi}^k\boldsymbol{\xi}^l}^{\gamma}} - \frac{q_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}^{(\gamma+1)}}{\sum_{kl} q_{\boldsymbol{\xi}^k\boldsymbol{\xi}^l}^{(\gamma+1)}}\right)$
$\mathrm{D}_{\mathrm{CS}}(p\|q)$ Eq. (9.30)	$\frac{\partial \mathrm{D}_{\mathrm{CS}}}{\partial \boldsymbol{\xi}} = 4 \int \frac{\boldsymbol{\xi}-\boldsymbol{\zeta}}{1+r} \left(\frac{p\, q}{\int p' q' \, dr'} - \frac{q^2}{\int q'^2 dr'}\right) d\boldsymbol{\zeta}$	$\frac{\partial \mathrm{D}_{\mathrm{CS}}}{\partial \boldsymbol{\xi}^i} = 4 \sum_j^n \frac{\boldsymbol{\xi}^i-\boldsymbol{\xi}^j}{1+r_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}} \left(\frac{p_{\boldsymbol{x}^i\boldsymbol{x}^j}\; q_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}}{\sum_{kl} p_{\boldsymbol{x}^k\boldsymbol{x}^l} q_{\boldsymbol{\xi}^k\boldsymbol{\xi}^l}} - \frac{q_{\boldsymbol{\xi}^i\boldsymbol{\xi}^j}^2}{\sum_{kl} q_{\boldsymbol{\xi}^k\boldsymbol{\xi}^l}^2}\right)$

Once again, we use Eq. (9.56) to calculate the gradient of D_γ with respect to $\boldsymbol{\xi}$:

$$\begin{aligned}
\frac{\partial \mathrm{D}_\gamma}{\partial \boldsymbol{\xi}} =& \frac{-4}{Q_\gamma V_\gamma} \int \frac{q(\boldsymbol{\xi}-\boldsymbol{\zeta})}{1+r} \Bigg(q^\gamma V_\gamma - p q^{(\gamma-1)} Q_\gamma - \\
& \qquad \int \left(q'^\gamma V_\gamma - p'\, q'^{(\gamma-1)} Q_\gamma \right) q' \Pi_{r'}\, dr' \Bigg) d\boldsymbol{\zeta} \\
=& -\frac{4}{Q_\gamma V_\gamma} \int \frac{q(\boldsymbol{\xi}-\boldsymbol{\zeta})}{1+r} \Bigg(q^\gamma V_\gamma - p\, q^{(\gamma-1)} Q_\gamma - V_\gamma \\
& \qquad \int q'^{(\gamma+1)} \Pi_{r'}\, dr' + Q_\gamma \int p' q'^\gamma \Pi_{r'}\, dr' \Bigg) d\boldsymbol{\zeta} \\
=& -\frac{4}{Q_\gamma V_\gamma} \int \frac{q(\boldsymbol{\xi}-\boldsymbol{\zeta})}{1+r} \left(q^\gamma V_\gamma - p\, q^{\gamma-1} Q_\gamma - V_\gamma Q_\gamma + Q_\gamma V_\gamma \right) d\boldsymbol{\zeta} \\
=& 4 \int \frac{\boldsymbol{\xi}-\boldsymbol{\zeta}}{1+r} \left(\frac{p\, q^\gamma}{\int p' q'^\gamma dr'} - \frac{q^{(\gamma+1)}}{\int q'^{(\gamma+1)} dr'} \right) d\boldsymbol{\zeta}.
\end{aligned} \tag{9.69}$$

For the special choice $\gamma = 1$ the Gamma-divergence becomes the Cauchy-Schwarz divergence Eq. (9.30) and the gradient $\frac{\partial \mathrm{D}_{\mathrm{CS}}}{\partial \boldsymbol{\xi}}$ for t-SNE can be directly derived from Eq. (9.69):

$$\frac{\partial \mathrm{D}_{\mathrm{CS}}}{\partial \boldsymbol{\xi}} = 4 \int \frac{\boldsymbol{\xi}-\boldsymbol{\zeta}}{1+r} \left(\frac{p\, q}{\int p' q'\, dr'} - \frac{q^2}{\int q'^2 dr'} \right) d\boldsymbol{\zeta} \ . \tag{9.70}$$

Moreover, similar derivations can be made for any other divergence, since one only needs to calculate the Fréchet-derivative of the divergence and apply it to Eq. (9.56).

9.6 SONE using arbitrary divergences

Similar to the SNE and t-SNE methods, also the SONE (see Algorithm 8.2) can be generalized to employ different divergences. Based on the special case of the GKL divergence employed in Eq. (8.10) we define a cost function for arbitrary Divergences $\mathrm{D}(p\|q)$:

$$E_{\mathrm{SONE}} = \int \sum_i \delta_{\Psi_{\mathrm{D}}(\boldsymbol{s}),\boldsymbol{x}^i} \cdot \sum_j \mathrm{D}\left(h_\sigma^{\Psi_{\mathrm{D}}(\boldsymbol{s})}(j) \middle\| g_\varsigma^{\boldsymbol{s}}(j) \right) \, p(\boldsymbol{s}) d\boldsymbol{s} \ , \tag{9.71}$$

where the best matching data point $\Psi^{\mathrm{D}}(\boldsymbol{s})$ for $\boldsymbol{s}$ is defined as:

$$\Psi_{\mathrm{D}}(\boldsymbol{s}) = \boldsymbol{x}^i \text{ such that } \sum_j \mathrm{D}\left(h_\sigma^{\Psi_{\mathrm{D}}(\boldsymbol{s})}(j) \middle\| g_\varsigma^{\boldsymbol{s}}(j) \right) \text{ is minimum.} \tag{9.72}$$

Table 9.3: Summary of the SONE learning rules for positive measures and different divergences.

Divergence	learning rules Gaussian $g_\varsigma^{\boldsymbol{s}}$ Eq. (8.8)	learning rules t-distributed $g_\varsigma^{\boldsymbol{s}}$ Eq. (8.9)
gen. KL	$\Delta\boldsymbol{\xi}^k = \frac{1}{2\varsigma^2}\left(h_\sigma^{\Psi_{\mathrm{GKL}}(\boldsymbol{s})}(k) - g_\varsigma^{\boldsymbol{s}}(k)\right)\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)}{\partial \boldsymbol{\xi}^k}$	$\Delta\boldsymbol{\xi}^k = \frac{\varsigma+1}{2\varsigma}\frac{1}{(1+d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)/\varsigma)}\left(h_\sigma^{\Psi_{\mathrm{GKL}}(\boldsymbol{s})}(k) - g_\varsigma^{\boldsymbol{s}}(k)\right)\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)}{\partial \boldsymbol{\xi}^k}$
Itakura-Saito	$\Delta\boldsymbol{\xi}^k = \frac{1}{2\varsigma^2}\left(\frac{h_\sigma^{\Psi_{\mathrm{IS}}(\boldsymbol{s})}(k)}{g_\varsigma^{\boldsymbol{s}}(k)} - 1\right)\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)}{\partial \boldsymbol{\xi}^k}$	$\Delta\boldsymbol{\xi}^k = \frac{\varsigma+1}{2\varsigma}\frac{1}{(1+d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)/\varsigma)}\left(\frac{h_\sigma^{\Psi_{\mathrm{IS}}(\boldsymbol{s})}(k)}{g_\varsigma^{\boldsymbol{s}}(k)} - 1\right)\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)}{\partial \boldsymbol{\xi}^k}$
Beta-div.	$\Delta\boldsymbol{\xi}^k = \frac{1}{2\varsigma^2}\cdot g_\varsigma^{\boldsymbol{s}}(k)^{(\beta-1)}\left(h_\sigma^{\Psi_\beta(\boldsymbol{s})}(k) - g_\varsigma^{\boldsymbol{s}}(k)\right)\cdot \frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)}{\partial \boldsymbol{\xi}^k}$	$\Delta\boldsymbol{\xi}^k = \frac{\varsigma+1}{2}\frac{g_\varsigma^{\boldsymbol{s}}(k)^{(\beta-1)}}{\varsigma+d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)}\left(h_\sigma^{\Psi_\beta(\boldsymbol{s})}(k) - g_\varsigma^{\boldsymbol{s}}(k)\right)\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)}{\partial \boldsymbol{\xi}^k}$
Alpha-div.	$\Delta\boldsymbol{\xi}^k = \frac{1}{2\varsigma^2}\cdot\frac{g_\varsigma^{\boldsymbol{s}}(k)}{\alpha}\left(\left(\frac{h_\sigma^{\Psi_\alpha(\boldsymbol{s})}(k)}{g_\varsigma^{\boldsymbol{s}}(k)}\right)^\alpha - 1\right)\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)}{\partial \boldsymbol{\xi}^k}$	$\Delta\boldsymbol{\xi}^k = \frac{\varsigma+1}{2\varsigma}\frac{1}{(1+d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)/\varsigma)}\frac{g_\varsigma^{\boldsymbol{s}}(k)}{\alpha}\left(\left(\frac{h_\sigma^{\Psi_\alpha(\boldsymbol{s})}(k)}{g_\varsigma^{\boldsymbol{s}}(k)}\right)^\alpha - 1\right)\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)}{\partial \boldsymbol{\xi}^k}$
Gamma-div.	$\Delta\boldsymbol{\xi}^k = -\frac{\left[g_\varsigma^{\boldsymbol{s}}(k)\right]^\gamma}{2\varsigma^2}\left(\frac{g_\varsigma^{\boldsymbol{s}}(k)}{\sum_j\left[g_\varsigma^{\boldsymbol{s}}(j)\right]^{\gamma+1}} - \frac{h_\sigma^{\Psi_\gamma(\boldsymbol{s})}(k)}{\sum_j h_\sigma^{\Psi_\gamma(\boldsymbol{s})}(j)\cdot\left[g_\varsigma^{\boldsymbol{s}}(j)\right]^\gamma}\right)\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)}{\partial \boldsymbol{\xi}^k}$	$\Delta\boldsymbol{\xi}^k = \frac{-(\varsigma+1)\cdot\left[g_\varsigma^{\boldsymbol{s}}(k)\right]^\gamma}{2(\varsigma+d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k))}\left(\frac{g_\varsigma^{\boldsymbol{s}}(k)}{\sum_j\left[g_\varsigma^{\boldsymbol{s}}(j)\right]^{\gamma+1}} - \frac{h_\sigma^{\Psi_\gamma(\boldsymbol{s})}(k)}{\sum_j h_\sigma^{\Psi_\gamma(\boldsymbol{s})}(j)\left[g_\varsigma^{\boldsymbol{s}}(j)\right]^\gamma}\right)\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)}{\partial \boldsymbol{\xi}^k}$
Cauchy-Schwarz	$\Delta\boldsymbol{\xi}^k = \left(\frac{g_\varsigma^{\boldsymbol{s}}(k)}{\sum_j\left[g_\varsigma^{\boldsymbol{s}}(j)\right]^2} - \frac{h_\sigma^{\Psi_{\mathrm{CS}}(\boldsymbol{s})}(k)}{\sum_j h_\sigma^{\Psi_{\mathrm{CS}}(\boldsymbol{s})}(j)\cdot g_\varsigma^{\boldsymbol{s}}(j)}\right)\cdot\left(-\frac{g_\varsigma^{\boldsymbol{s}}(k)}{2\varsigma^2}\right)\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)}{\partial \boldsymbol{\xi}^k}$	$\Delta\boldsymbol{\xi}^k = -\frac{(\varsigma+1)\cdot g_\varsigma^{\boldsymbol{s}}(k)}{2(\varsigma+d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k))}\left(\frac{g_\varsigma^{\boldsymbol{s}}(k)}{\sum_j\left[g_\varsigma^{\boldsymbol{s}}(j)\right]^2} - \frac{h_\sigma^{\Psi_{\mathrm{CS}}(\boldsymbol{s})}(k)}{\sum_j h_\sigma^{\Psi_{\mathrm{CS}}(\boldsymbol{s})}(j)\cdot g_\varsigma^{\boldsymbol{s}}(j)}\right)\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{\xi}^k)}{\partial \boldsymbol{\xi}^k}$

Here, we reused the definitions of Chapter 8, i.e. the Kronecker delta $\delta_{\cdot,\cdot}$, the sampling vectors $\boldsymbol{s}$, the neighborhood cooperation in the original space $h_\sigma^{\Psi_{\mathrm{D}}(\boldsymbol{s})}(j)$ Eq. (8.7) and the low-dimensional space $g_\varsigma^{\boldsymbol{s}}(j)$ Eqs. (8.8) or (8.9).

The derivative of the cost function (9.71) with respect to the image vectors $\boldsymbol{\xi}^k$ can be done using the Fréchet derivative Eq. (9.37):

$$\frac{\partial E_{\mathrm{SONE}}}{\partial \boldsymbol{\xi}^k} = \int \left[\frac{\delta \mathrm{D}\left(h_\sigma^{\Psi_{\mathrm{D}}(\boldsymbol{s})} \middle\| g_\varsigma^{\boldsymbol{s}}\right)}{\delta g_\varsigma^{\boldsymbol{s}}}[l] \cdot \frac{\partial g_\varsigma^{\boldsymbol{s}}}{\partial \boldsymbol{\xi}^k} \right] dl \tag{9.73}$$

$$= \int \left[\left. \frac{\delta \mathrm{D}\left(h_\sigma^{\Psi_{\mathrm{D}}(\boldsymbol{s})} \middle\| g_\varsigma^{\boldsymbol{s}}\right)}{\delta g_\varsigma^{\boldsymbol{s}}} \right|_l \cdot \delta_{l,k} \cdot \frac{\partial g_\varsigma^{\boldsymbol{s}}}{\partial \boldsymbol{\xi}^k} \right] dl \tag{9.74}$$

$$= \left. \frac{\delta \mathrm{D}\left(h_\sigma^{\Psi_{\mathrm{D}}(\boldsymbol{s})} \middle\| g_\varsigma^{\boldsymbol{s}}\right)}{\delta g_\varsigma^{\boldsymbol{s}}} \right|_k \cdot \frac{\partial g_\varsigma^{\boldsymbol{s}}(k)}{\partial \boldsymbol{\xi}^k} \ . \tag{9.75}$$

This yields the online learning update rule for a given sampling vector $\boldsymbol{s}$ and learning rate τ:

$$\boldsymbol{\xi}^k = \boldsymbol{\xi}^k - \tau \cdot \frac{\partial E_{\mathrm{SONE}}}{\partial \boldsymbol{\xi}^k} = \boldsymbol{\xi}^k - \tau \Delta \boldsymbol{\xi}^k \ . \tag{9.76}$$

Since the Fréchet derivatives of a wide selection of divergences is investigated in previous sections we can immediately write down learning rules for all divergence families. The explicit formulas in case of Gaussian and t-distributed neighborhood function $g_\varsigma^{\boldsymbol{s}}$ and different divergences can be found in Table 9.3.

9.7 Experiments

9.7.1 t-SNE incorporating Gamma-divergence vs. original t-SNE

In this section we demonstrate the applicability of the Gamma-divergence in the t-SNE method on the real world examples, namely the Olivetti faces [1] and the COIL-20 data set (Nene et al. 1996). The Olivetti data set consists of intensity-value pictures of 40 individuals with small variations in viewpoint, large variation in expression and occasional addition of glasses. The data set contains 400 images (10 per person) of size 64×64. The COIL-20 data set contains images of 20 different objects viewed from 72 equally spaced orientations. In total we have 1,440 images of $32 \times 32 = 1,024$ pixels. Like suggested in (van der Maaten and Hinton 2008) we

[1] The Olivetti faces data set is publicly available from http://cs.nyu.edu/~roweis/data.html

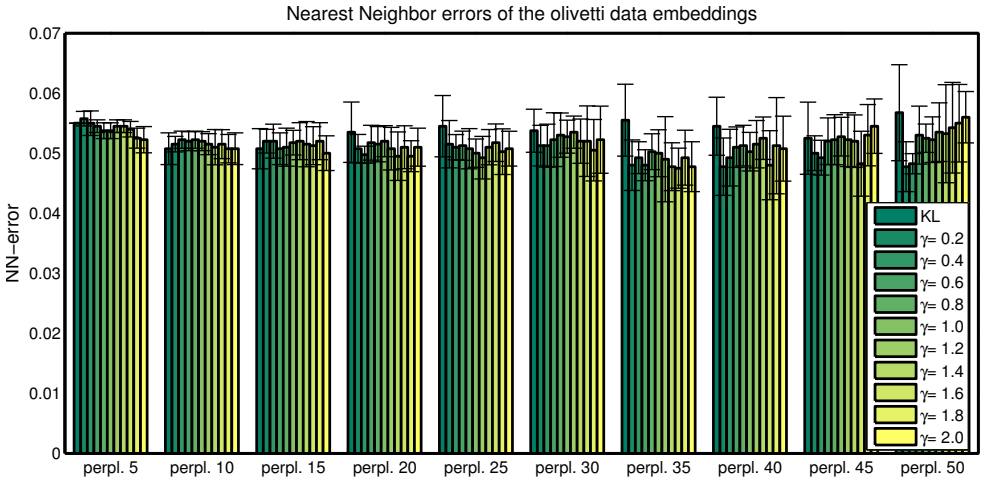


Figure 9.14: 1-NN errors of the 2 dim. Olivetti faces embeddings using the Gamma-divergence in comparison with KL for different perplexities.

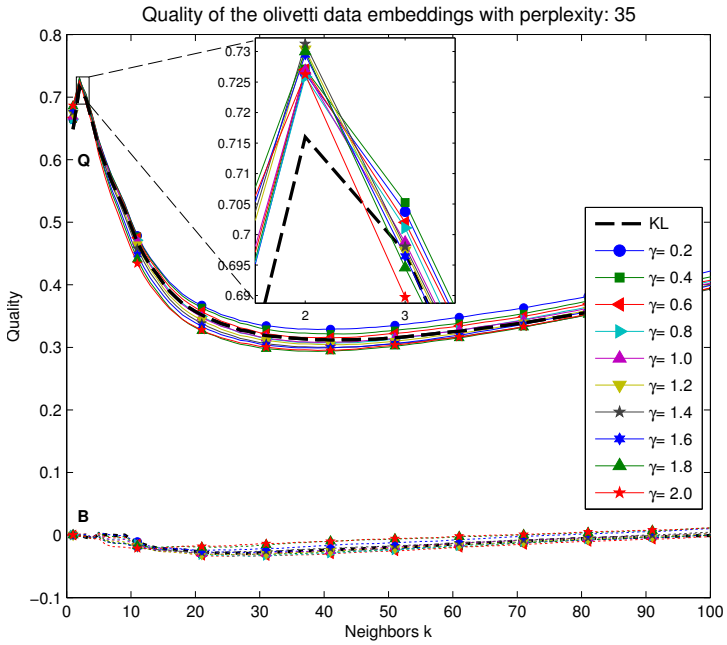


Figure 9.15: Quality of the 2 dim. embeddings using the Gamma-divergence on the Olivetti faces data in comparison with the original formulation using KL.

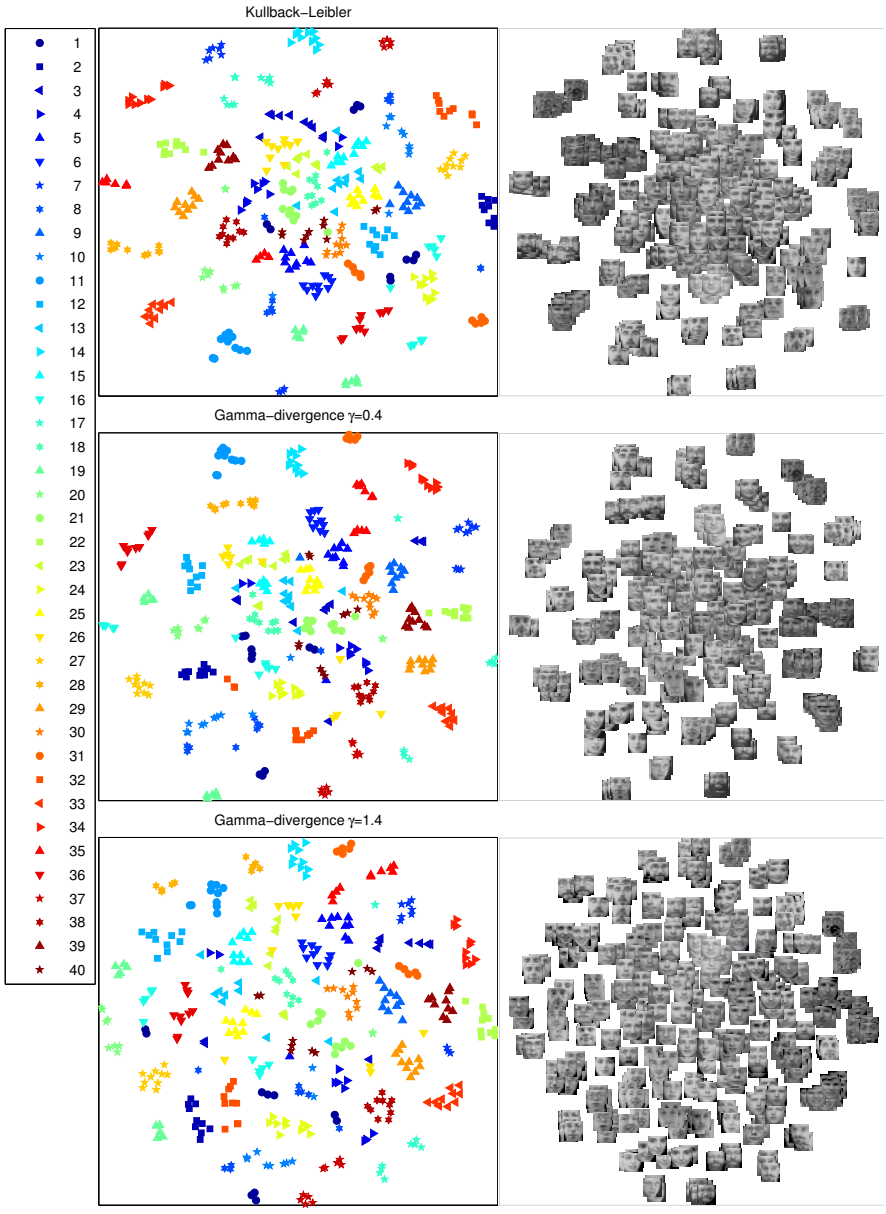


Figure 9.16: Embeddings of the Olivetti faces based on the same initialization for different divergences and perplexity 35.

preprocessed the data by extracting the mean and reducing the dimension to 30 using PCA and successive transformation to unit variance features. We constructed 10 independent random initializations for the experiments, which we reused in the algorithm with different divergences and values of the divergence parameter. To compare the different embeddings we use the 1-NN classification error using the persons as labels. A quantitative evaluation based on the quality measure as proposed by (Lee and Verleysen 2008, Lee and Verleysen 2009) (see Algorithm 6.6) is included.

Figure 9.14 shows the nearest neighbor errors of the embeddings of the Olivetti data as mean and standard deviation over the 10 random initializations for different perplexities and Gamma-divergences with γ varying in the interval $[0.2\ 2]$. Dependent on the perplexity the influence of the divergence varies. For small perplexities, greater values of γ show better classification accuracy, while for large perplexities lower γ yield better performance. Nevertheless, in this data set the use of the Gamma-divergence leads, in most cases, to a slight improvement of the nearest neighbor classification compared to the Kullback-Leibler divergence.

Figure 9.15 shows the quantitative evaluation on Olivetti using the intrusion- and extrusion measure mentioned above as mean over the 10 random initializations in the example case of perplexity 35. Again we observe small deviations in the behavior depending on the choice of the divergence. Some example visualizations are shown in Figure 9.16. For comparison all visualizations are based on the same initialization. Note that, for example, the data points representing person 35 are widely scattered in the embedding space when using the Kullback-Leibler divergence, while they remain close together when using the Gamma-divergence.

Figure 9.17 shows the 1-NN errors of the embeddings for COIL-20 as a mean and standard deviation over the 10 random initializations for different perplexities and Gamma-divergences with γ varying in the interval $[0.2\ 2]$. Dependent on the perplexity the influence of the divergence varies. For small perplexities error free visualizations are possible in all cases. For big perplexities in this data set the usage of the Gamma-divergence leads to an improvement of the 1-NN classification in comparison with Kullback-Leibler. Furthermore, it is clearly visible that the Gamma-divergence is quite robust for $\gamma > 0.4$ in this case.

Figure 9.18 shows the quantitative evaluation using the intrusion- and extrusion measure (see Algorithm 6.6) as mean over the 10 random initializations in the example case of perplexity 25. Again we observe small deviations in the behavior dependent on the choice of the divergence. Some example visualizations based on the same initialization are shown in Figure 9.19. Note that, for example, the data points representing object 1 are chained on a bended line using the Kullback-Leibler divergence, while it is visualized in a closed loop using the Gamma-divergence with

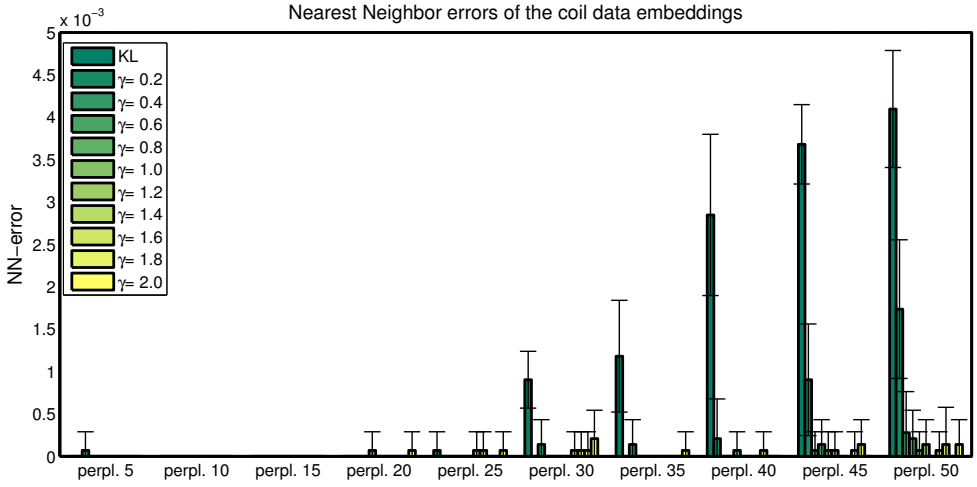


Figure 9.17: 1-NN errors of the 2 dim. COIL-20 embeddings using the Gamma-divergence in comparison with KL for different perplexities.

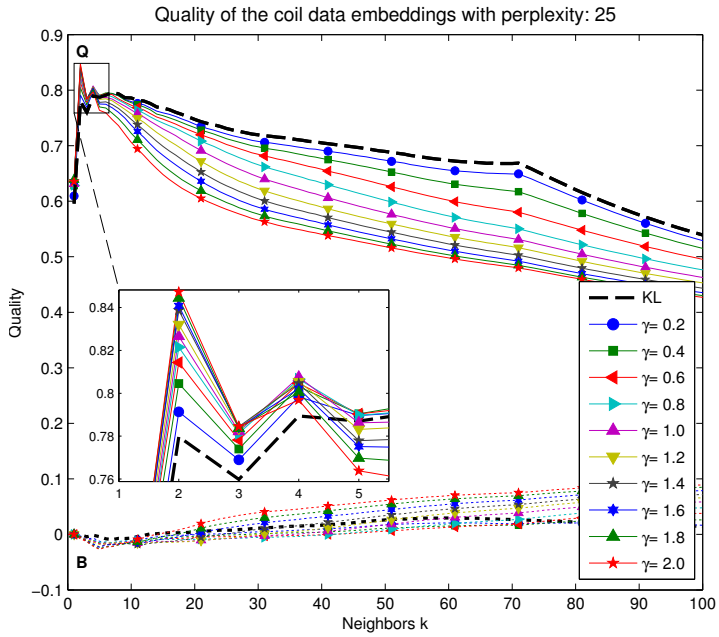


Figure 9.18: Quality of the 2 dim. embeddings using the Gamma-divergence on the COIL-20 data in comparison with the original formulation using KL.

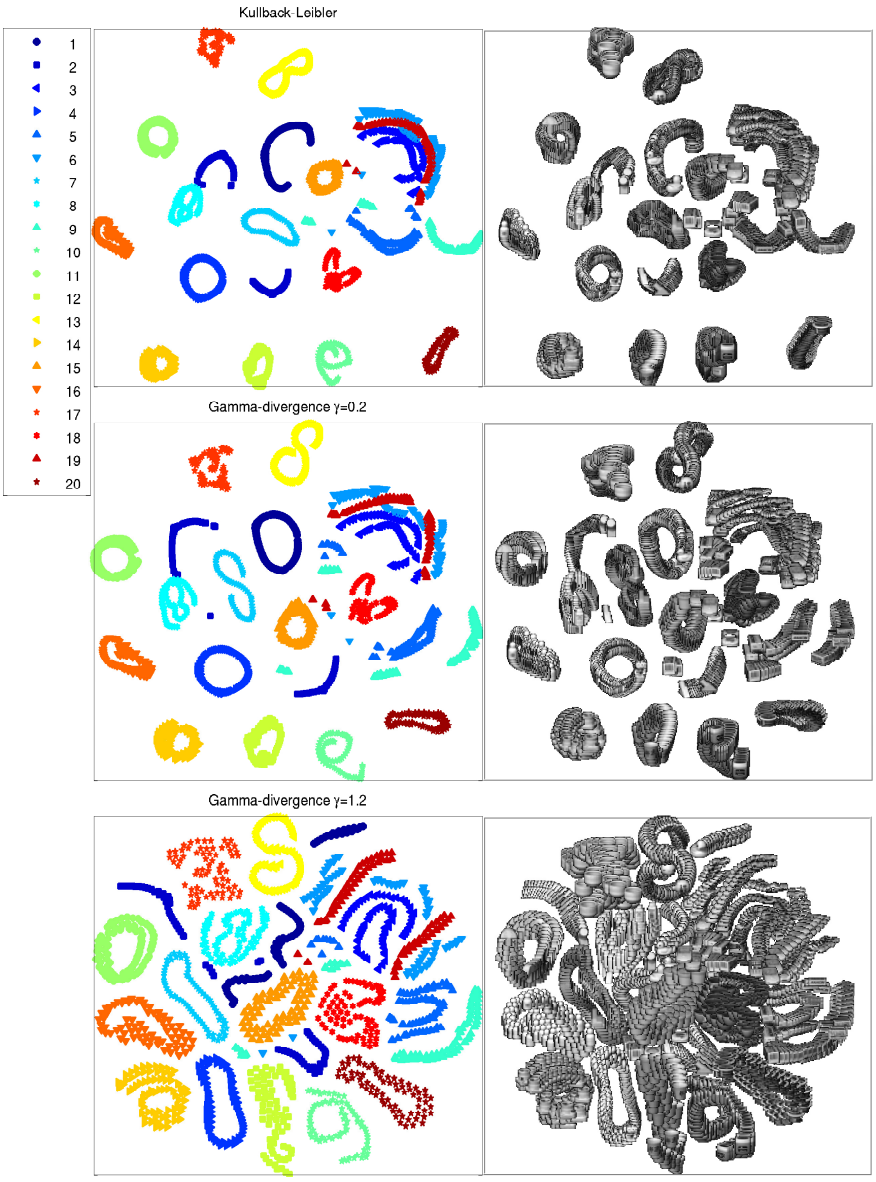


Figure 9.19: Embeddings of the COIL-20 data set based on the same initialization for different divergences and perplexity 25.

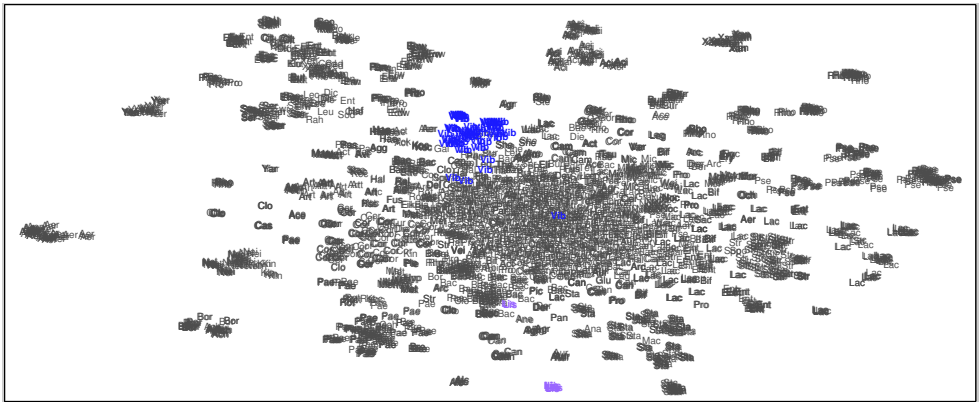


Figure 9.20: Best t-SNE similarity map of the Bacteria data set using perplexity 15.

$\gamma = 0.2$. For bigger values of γ the quality of embeddings with respect to small neighborhoods increases in comparison to the original formulation using KL. From visual inspection one observes, that the maps show more local details comparable to the similarity maps of SONE in Chapter 8. This comes at the cost of loosing quality for bigger neighborhoods, i.e. some global aspects might get lost. It can be seen for $\gamma = 1.2$, where the chains of object 1 and 19 completely brake.

9.7.2 Bacteria similarity map generated by SONE

The identification of bacteria is an important task in medicine or biology and is often done using large data bases with reference signatures (Maier et al. 2006). The reference spectra of the different bacteria species are in parts very similar and multimodal as an additional challenge for the identification methods. To maintain these data bases efficient exploration and visualization tools are necessary. Common tasks are the identification of outliers, strong overlapping and therefore hard to distinguish data clusters or erroneous measurements.

Here we consider a database of $n = 3048$ bacteria samples measured and prepared in accordance to (Barbuddhe et al. 2008, Maier et al. 2006). Each sample is given as a vector $\boldsymbol{x} \in \mathbb{R}^N$, with dimensionality N (number of peaks), considered as a function p. Overall the data contain around 200 species in accordance to the taxonomy of bacteria and are quite challenging for visualization. For each $\boldsymbol{x}$ a labeling is available in the following abbreviated by a three letter code. The map obtained with original t-SNE (Fig. 9.20) is able to separate some clusters of bacteria, but the center is more crowded than the SONE map, see Fig. 9.21. The SONE embedding was ob-

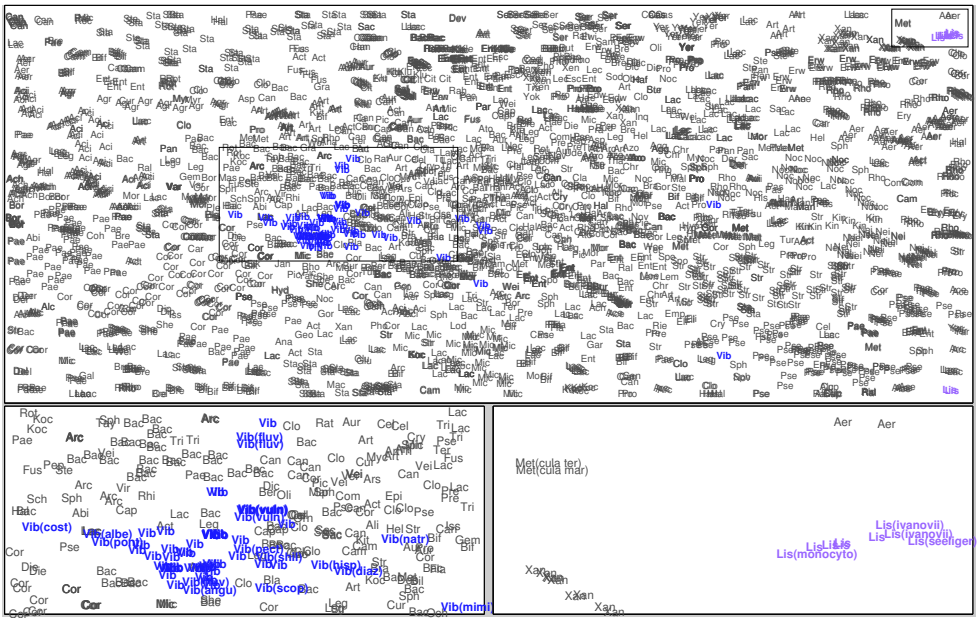


Figure 9.21: SONE embedding of the Bacteria data set and two zoomed regions.

tained using uniformly distributed sampling vectors $\boldsymbol{s}$. The KL divergence was used to spread the samples globally on the map. Afterwards we trained for further 100 epochs using the Gamma-divergence with $\gamma = 0.5$, which controls the granularity and results in higher quality for small neighborhoods. In contrast to other methods SONE enforce spreading of the data samples on the given structure hypothesis and allows to influence the granularity, which enhances visibility of single samples. The quality of both the SONE and t-SNE embedding measured by intrusions and extrusions behaves quite similar for this data set.

The SONE representation was already quite effective in representing the many bacteria spectra and similar samples are indeed plotted near to each other, which is in good agreement to the expectations of the experts (Maier et al. 2006). The map also allows to identify isolated clusters like the one depicted in the right zoomed regions of Fig. 9.21. This plot contains most of the *Listeria* spectra from the database which are known to be very distinctive. For the second zoomed region (left) a large cohort of *Vibrio* spectra is shown. It is more diverse and very well represented, but we can also identify more distant Vibrio items which by closer inspection are indeed special cases. The map allows the biochemical expert to navigate through the similarity space and to analyze spectra found to be (dis-)similar by the model.

9.8 Conclusion and outlook

The original SNE, t-SNE and SONE formulation employ the Kullback-Leibler divergence to measure the disagreement of the topology in the high- and low-dimensional space, respectively. In this Chapter we provide a mathematical foundation for the use of arbitrary divergences and their derivatives such that they can immediately be plugged into the existing algorithms. This provides the reader with alternative measures, which can be used if the results using Kullback-Leibler are not satisfying.

Therefore, we characterize main subclasses of divergences following (Cichocki et al. 2009): Bregman-, Csiszár f- and Gamma-divergences. We used the mathematical methodology of Fréchet derivatives to obtain the generalized gradients for the methods. And we derived the t-SNE and SONE gradients for a wide range of important divergences as summarized in Table 9.2 and Table 9.3.

We studied the behavior of the divergences in some experiments inspired by image processing. From the experiments it is clearly visible that the divergences show different behavior for different problems. Although we are not yet able to deliver an overall recipe for chosing a particular divergence in a given task, we can still argue that it might be advantageous to try alternative measures if the results are not satisfying. As an example, we discuss the t-SNE method using the Gamma-divergence, considering the publicly available Olivetti faces and COIL-20 data sets. Performances are compared in terms of the 1-NN classification error of the embeddings, the quality as measured by intrusion- and extrusion behavior (Lee and Verleysen 2008, Lee and Verleysen 2009) and by visual inspection. The extension of SONE is illustrated by means of a similarity map in the domain of Bacteria diversity.

The investigation of further divergences on more data sets will be addressed in further studies. Furthermore, divergences like Alpha-, Beta-, Eta-, Gamma-, generalized Rényi, and generalized Kullback-Leibler divergence do not require probability densities as inputs, but can be applied to positive measures. Through normalization information might get lost, so the use of generalized divergences on non-normalized neighborhood functions for SNE and t-SNE improves performances, potentially. This will be investigated in forthcoming projects.

9.A Derivative of the general t-SNE gradient

In this Section we derive the general form of the t-SNE gradient using the definitions introduced in Section 9.4.1. Furthermore, we will abbreviate Eq. (9.54) for reasons of clarity as

$$q(r') = f\left(r'\right) \cdot I^{-1} \ . \tag{9.77}$$

Let us consider the derivative of D with respect to $\boldsymbol{\xi}$:

$$\begin{aligned}\frac{\partial \mathrm{D}}{\partial \boldsymbol{\xi}} &= \frac{\partial \mathrm{D}\left(p, q(r(\boldsymbol{\xi}, \boldsymbol{\zeta}))\right)}{\partial \boldsymbol{\xi}} = \int\int \frac{\delta \mathrm{D}}{\delta r'} \frac{\partial r'}{\partial \boldsymbol{\xi}} \, d\boldsymbol{\xi}' d\boldsymbol{\zeta}' = \int\int \frac{\delta \mathrm{D}}{\delta r\left(\boldsymbol{\xi}', \boldsymbol{\zeta}'\right)} \frac{\partial r\left(\boldsymbol{\xi}', \boldsymbol{\zeta}'\right)}{\partial \boldsymbol{\xi}} \, d\boldsymbol{\xi}' d\boldsymbol{\zeta}' \\ &= 4 \int \frac{\delta \mathrm{D}}{\delta r\left(\boldsymbol{\xi}, \boldsymbol{\zeta}\right)} \left(\boldsymbol{\xi} - \boldsymbol{\zeta}\right) \, d\boldsymbol{\zeta} \ .\end{aligned} \tag{9.78}$$

Using the chain rule for functional derivatives we get:

$$\frac{\delta \mathrm{D}}{\delta r\left(\boldsymbol{\xi}, \boldsymbol{\zeta}\right)} = \int\int \frac{\delta \mathrm{D}}{\delta q(r(\boldsymbol{\xi}', \boldsymbol{\zeta}'))} \frac{\delta q(r(\boldsymbol{\xi}', \boldsymbol{\zeta}'))}{\delta r\left(\boldsymbol{\xi}, \boldsymbol{\zeta}\right)} d\boldsymbol{\xi}' d\boldsymbol{\zeta}' = \int \frac{\delta \mathrm{D}}{\delta q(r')} \frac{\delta q(r')}{\delta r} \Pi_{r'} \, dr' \tag{9.79}$$

where

$$\begin{aligned}\frac{\delta q(r')}{\delta r} &= \frac{\delta f\left(r'\right)}{\delta r} \cdot I^{-1} - f(r') \cdot I^{-2} \frac{\delta I}{\delta r} \quad \text{holds, with} \\ \frac{\delta f(r')}{\delta r} &= -\delta_{r,r'} \left(1 + r\right)^{-2} \text{ and } \frac{\delta I}{\delta r} = -\left(1 + r\right)^{-2} .\end{aligned}$$

So we obtain

$$\begin{aligned}\frac{\delta q(r')}{\delta r} &= f\left(r'\right) \cdot I^{-2} \cdot \frac{1}{\left(1 + r\right)^2} - \frac{\delta_{r,r'} \left(1 + r\right)^{-2}}{I} \\ &= \frac{f\left(r'\right)}{I} \frac{f\left(r\right)}{I} \frac{1}{\left(1 + r\right)} - \frac{\delta_{r,r'} \left(1 + r\right)^{-1} f\left(r\right)}{I} \\ &= q(r')q(r) \frac{1}{\left(1 + r\right)} - \delta_{r,r'} \left(1 + r\right)^{-1} q(r) \\ &= -\left(1 + r\right)^{-1} q(r) \left(\delta_{r,r'} - q(r')\right) .\end{aligned}$$

Substituting these results in Eq. (9.79), we get

$$\begin{aligned}\frac{\delta \mathrm{D}}{\delta r} &= \int \frac{\delta D}{\delta q(r')} \frac{\delta q(r')}{\delta r} \Pi_{r'} \, dr' = -\frac{q(r)}{1 + r} \int \frac{\delta \mathrm{D}}{\delta q(r')} \left(\delta_{r,r'} - q(r')\right) \Pi_{r'} \, dr' \\ &= -\frac{q(r)}{1 + r} \left(\frac{\delta \mathrm{D}}{\delta q(r)} - \int \frac{\delta \mathrm{D}}{\delta q(r')} q(r') \Pi_{r'} \, dr' \right) \ .\end{aligned}$$

The collection of all terms lead to the general derivative $\frac{\partial \mathrm{D}}{\partial \boldsymbol{\xi}}$ Eq. (9.56).

9.B Derivative of the general SNE gradient

Based on the definitions of Section 9.4.2 we derive the general formulation of the SNE gradient for arbitrary divergences. For the computation of the Fréchet derivative we can use the results from above for t-SNE. The only term that differs is the derivative of $q_{\mathrm{SNE}}\left(r'\right)$ with respect to r. For reasons of clarity we abbreviate q_{SNE} Eq. (9.57) by:

$$q_{\mathrm{SNE}}\left(r'\right) = g\left(r'\right) \cdot J^{-1} \quad . \tag{9.80}$$

We get

$$\frac{\delta q_{\mathrm{SNE}}\left(r'\right)}{\delta r} = \frac{\delta g\left(r'\right)}{\delta r} \cdot J^{-1} - g\left(r'\right) \cdot J^{-2} \frac{\delta J}{\delta r}$$

with

$$\frac{\delta g\left(r'\right)}{\delta r} = -\delta_{r,r'} \exp\left(-r\right) \text{ and } \frac{\delta J}{\delta r} = -\exp\left(-r\right) \quad ,$$

which leads to

$$\begin{aligned}
\frac{\delta q_{\mathrm{SNE}}\left(r'\right)}{\delta r} &= \frac{-\delta_{r,r'} \exp\left(-r\right)}{J} + g\left(r'\right) J^{-2} \exp\left(-r\right) \\
&= \frac{-\delta_{r,r'} g\left(r\right)}{J} + \frac{g\left(r'\right)}{J} \frac{g\left(r\right)}{J} = -\delta_{r,r'} q_{\mathrm{SNE}}\left(r\right) + q_{\mathrm{SNE}}\left(r'\right) q_{\mathrm{SNE}}\left(r\right) \\
&= -q_{\mathrm{SNE}}\left(r\right) \left(\delta_{r,r'} - q_{\mathrm{SNE}}\left(r'\right)\right) \quad .
\end{aligned}$$

Substituting these results in Eq. (9.79), we get

$$\begin{aligned}
\frac{\delta \mathrm{D}}{\delta r} &= \int \frac{\delta \mathrm{D}}{\delta q_{\mathrm{SNE}}\left(r\prime\right)} \frac{\delta q_{\mathrm{SNE}}\left(r'\right)}{\delta r} \Pi_{r'} \, dr' \\
&= -q_{\mathrm{SNE}}\left(r\right) \cdot \int \frac{\delta \mathrm{D}}{\delta q_{\mathrm{SNE}}\left(r\prime\right)} \left(\delta_{r,r'} - q_{\mathrm{SNE}}\left(r'\right)\right) \Pi_{r'} \, dr' \\
&= -q_{\mathrm{SNE}}\left(r\right) \cdot \left[\frac{\delta \mathrm{D}}{\delta q_{\mathrm{SNE}}\left(r\right)} - \int \frac{\delta \mathrm{D}}{\delta q_{\mathrm{SNE}}\left(r'\right)} q_{\mathrm{SNE}}\left(r'\right) \Pi_{r'} dr'\right] \quad .
\end{aligned}$$

The collection of all terms lead to the general derivative $\frac{\partial \mathrm{D}}{\partial \boldsymbol{\xi}}$ Eq. (9.58).

Chapter 10

Conclusion

Ideas are like rabbits. You get a couple and learn how to handle them, and pretty soon you have a dozen.

John Steinbeck (1902 - 1968)

10.1 Summary

This thesis presents several extensions of the Generalized LVQ (GLVQ) algorithm based on the concept of adaptive similarity measures. The metric learning gives rise to a variety of applications, including Content Based Image Retrieval (CBIR), supervised dimension reduction and advanced texture learning in image analysis, just to name a few. The detailed investigation of dimensionality reduction is addressed in the second half of the thesis. It includes the investigation of generalized explicit dimension reduction mappings for unsupervised and supervised dimension reduction. A novel technique for efficient unsupervised non-linear dimension reduction is proposed combining the concept of fast online learning and optimization of divergences. Finally, three divergence based algorithms are generalized and investigated for the use of arbitrary divergences.

In Chapter 2 the required background for adaptive metric learning and prototype-based classification is provided. Then, the Limited Rank Matrix LVQ (LiRaM LVQ) is introduced in Chapter 3, which aims at efficient optimization of classification especially for very high-dimensional data sets. By limiting the rank of the adaptive matrix, which is part of the used distance, the number of free parameters can be controlled explicitly. We show, that, besides the computational efficiency, limiting the rank shows superior quality in comparison to alternative approaches based on the eigenvalue decomposition after training, in particular if the target dimension is below the intrinsic dimensionality of the data set. Furthermore, this concept allows discriminant linear dimension reduction, aiming at the preservation of the classification accuracy in low dimensions. By decomposing the distance measure into global and local or class-wise matrices more complex decision boundaries can be realized into the visualization. This combines linear dimension reduction with

localized similarity measures in the low-dimensional space, defining non-linear decision boundaries of the receptive fields. The dimension reduction with LiRaM LVQ shows comparable or better results than alternative state-of-the-art techniques. Furthermore, the approach is also computationally efficient. In contrast to other high-quality techniques it does not require the computation of pair-wise affinities of the data points, but their distance with respect to the (few) prototypes, which typically accounts for much less computations. Several experiments on real-world data sets are presented and confirm our claims.

Chapter 4 presents an example application of the LiRaM LVQ in the context of CBIR. In many medical applications the amount of data is growing tremendously in recent years. Therefore, computer aided diagnosis systems, which automatically browse data bases and pre-select potentially interesting data for a given task are highly desirable. This work addresses CBIR in the context of Dermatology. In a joint project the Department of Dermatology of the University Medical Center Groningen provided an image data base with different types of skin lesions. The aim is to find a predefined number of similar pictures from the data base given a query image. With adaptive metrics we are able to increase the correct retrieval rates significantly for arbitrary color spaces. We compare two distance learning techniques: the Large Margin Nearest Neighbor (LMNN) and LiRaM LVQ approach. Interestingly, the LiRaM LVQ outperformed the LMNN based in sample settings. With growing complexity and time consumption of LMNN, similar results could be achieved.

In Chapter 5 we introduce a complex variant of GLVQ for texture classification, called Color Image Analysis LVQ (CIA LVQ). This flexible approach combines discriminative local linear projections in Fourier domain with linear filtering, e.g. with Gabor filters. Linear filtering operations are frequently defined on intensity values. Some heuristic techniques have been proposed for filter operations on color images combining responses or energies of color channels in some meaningful way. Our approach differs in nature, because it is based on an automatic learning procedure guided by supervised training. Therefore, a Gabor filter bank is a priori collected, using scales and orientations fitting the texture recognition task. We extract random patches from known classes of colored images and for each of them we transform the color channels separately into Fourier domain. The transformations of the color values to intensity values is learned by the CIA LVQ system optimizing the discrimination of the filter responses on these transformed patches. In particular for natural textures like bark and food structures, the proposed technique outperformed alternative approaches and the naive usage of an RGB to gray transformation, which is often used in practice. Furthermore, the CIA LVQ shows excellent generalization ability with respect to evaluation images which were never shown to the system before.

Part II of this thesis addresses different aspects concerning dimension reduction. In Chapter 6 a novel general view is proposed, which facilitates the adaptation of a variety of dimension reduction methods for explicit mappings. Instead of the implicit optimization of the positions of the low-dimensional data points we predefine the form of a mapping function f_W parameterized by W and optimize the parameters with respect to a specific objective. This has the advantage that the training can be performed on a small subset of the data only and a direct out-of-sample extension for all data points is immediately available. Furthermore, a theoretical investigation of the generalization ability for dimension reduction becomes possible. We demonstrate the concept of dimension reduction mappings based on the t-distributed SNE (t-SNE) cost function and different alternatives for the mapping function f_W. This includes unsupervised linear as well as non-linear mappings based on local PCA and supervised mappings using discriminative local linear projections. We compare the approach with several state-of-the art techniques, show the excellent generalization ability for several data sets and finally address the theoretical investigations of dimension reduction mappings. In all cases our approach displays comparable or even superior results.

Chapter 7 investigates supervised dimension reduction based on adaptive distances and local linear projections obtained by GMLVQ and LiRaM LVQ. It allows the integration of the dimension reduction into the optimization procedure aiming at discriminative visualizations. We show in terms of several examples that existing dimension reduction methods can be extended to a supervised setting using the learned metrics and discriminative transformations of LVQ.

In Chapter 8 an unsupervised dimension reduction technique is proposed, which combines fast sequential online learning and direct divergence optimization as used by SNE and t-SNE. The technique is called Self Organized Neighbor Embedding (SONE) and it exhibits several interesting properties: In its original formulation SONE is based on a structure hypothesis, which enables the user to control the appearance of the final embedding and adjust the computational effort. Many dimension reduction techniques require the computation of all pair-wise affinities of the low-dimensional image vectors during one optimization step. This leads to a computational complexity $\mathcal{O}(n^2)$, where n denotes the number of data points. SONE computes distances to one sampling vector drawn from the given hypothesis in each iteration for the adaptation of all points. Thus, the computational complexity is linearly dependent on the number of points and sampling vectors given by the hypothesis. Even though the method is less complex than SNE and t-SNE, it displays comparable quality as demonstrated in terms of several examples.

Chapter 9 addresses a systematic approach to the mathematical treatment of divergence based dimension reduction, such as SNE, t-SNE and SONE to exchange

their respective modules. Besides the independent treatment of the distribution in the low-dimensional space, e.g. using a Gaussian for SNE and a t-distribution in t-SNE, we concentrate on the divergence which measures the difference between distributions in the original and the embedding space. Therefore, we review the families of divergences and their properties. We propose a general framework based on the concept of Fréchet-derivatives and derive the explicit learning rules for a wide range of divergences. In the experiments we concentrate on the evaluation of the Gamma-divergence for t-SNE and SONE in several real-world data sets. We observed that the Gamma-divergence enhances the quality of the embeddings for small neighborhoods in comparison with the original formulation using Kullback-Leibler.

10.2 Future work

This work can be extended in several directions. Future projects may concern the enhancement of the computer aided diagnosis system based on CBIR. This could be achieved by the incorporation of more elaborate features. Furthermore, the CIA LVQ for classification as well as all proposed approaches regarding dimension reduction are constructed in a generic way, which allows for easy adaptation and exchange of the modules. This enables flexible customization with respect to the user specific needs and the desired application. In particular, we suggest the following topics for future research:

- Learning Vector Quantization (LVQ) has shown to be particularly attractive for interdisciplinary applications in medical or biological domains. Apart from the application of the proposed LVQ variants to other data sets, we intent to put forward the CBIR system for dermatological images introduced in Chapter 4. It is based on the most simple features for color images: the mean color values of lesion and healthy skin. Extensions may also take into account shape and texture information of lesions, e.g. using shape extraction methods and the CIA LVQ proposed in Chapter 5. Furthermore, color histograms could be extracted and investigated by divergence LVQ (Mwebaze et al. 2011). Moreover, the color classes of the lesions may be subdivided into more detailed disease classes leading more precise retrievals.

- The CIA LVQ as introduced in Chapter 5 needs an a priory defined filter bank as input. Similarly to Gabor filters any other family of 2D filters commonly used to describe gray scale image information could be adapted and applied to color image analysis with this algorithm. A filter bank with differences of

Gaussians for color edge detection is a possible example. The investigation of the performance of the system for other filters can be addressed in future.

Furthermore, depending on the task it might be desirable that two patches in which the same texture occurs on different positions should not be interpreted as similar. In this case another similarity measure should be used which is not based on the difference of magnitudes. This might be of advantage for example in the recognition of objects such as traffic signs, where a corner or an edge might have different meanings dependent on its position in the image.

Note, that the incorporation of prior knowledge in form of a predefined filter bank might not always be feasible. Actually, the algorithm theoretically allows the optimization with respect to all variables. Thus, it is possible to include the local filters into the optimization process. First experiments based on that concept showed already promising results. The investigation of this extension will be addressed in forthcoming projects.

- Obviously, the general framework introduced in Chapter 6 gives rise to the investigation of alternative dimension reduction mappings based on other cost functions and other functional forms of the mapping. Moreover the derivation of explicit bounds concerning the generalization ability may be the subject of future work.

  At present, the setting has been restricted to vectorial data only due to the form of the mapping f_W. Naturally, more general forms could be considered which can take more complex, non-vectorial data as inputs, such as mappings which are based on general dissimilarity characterization. A corresponding investigation will be the subject of forthcoming projects.

- Chapter 9 introduces the extension of divergence-based dimension reduction to a general framework using arbitrary divergences. The investigation of further divergences on more data sets could be addressed in further studies. Furthermore, various divergences including the generalized Kullback-Leibler divergence do not require probability densities as inputs, but can be applied to positive measures. Through normalization information might get lost, so the use of generalized divergences in non-normalized neighborhood functions for SNE and t-SNE improves performances, potentially. This can be investigated in future projects.

Publications

Journal Papers

[1] K. Bunte, M. Biehl, and B. Hammer. A general framework for dimensionality reducing data visualization using explicit mapping functions. *Neural Computation*, 2011. Accepted for publication.

[2] K. Bunte, P. Schneider, B. Hammer, Frank-Michael Schleif, T. Villmann, and M. Biehl. Limited rank matrix learning – discriminative dimension reduction and visualization. *Neural Networks*, 2011. Accepted for publication.

[3] K. Bunte, M. Biehl, M. F. Jonkman, and N. Petkov. Learning effective color features for content based image retrieval in dermatology. *Pattern Recognition*, 44(9):1892–1902, September 2011.

[4] K. Bunte, B. Hammer, T. Villmann, M. Biehl, and A. Wismüller. Neighbor embedding XOM for dimension reduction and visualization. *Neurocomputing*, 74(9):1340–1350, 2011.

[5] P. Schneider, K. Bunte, B. Hammer, and M. Biehl. Regularization in matrix relevance learning. *IEEE Transactions on Neural Networks*, 21(5):831–840, May 2010.

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

Conference Papers

[1] K. Bunte, I. Giotis, N. Petkov, and M. Biehl. Adaptive matrices for color texture classification. In *14th International Conference on Computer Analysis of Images and Patterns (CAIP)*, volume 6855 of *Lecture Notes in Computer Science*, pages 489–497, Seville, Spain, August 2011.

[2] B. Hammer, M. Biehl, K. Bunte, and Bassam Mokbel. A general framework for dimensionality reduction for large data sets. In Jorma Laaksonen and Timo Honkela, editors, *Advances in Self-Organizing Maps, WSOM 2011*, Lecture Notes in Computer Science 6731, pages 277–287. Springer, 2011.

[3] K. Bunte, F.-M. Schleif, S. Haase, and T. Villmann. Mathematical foundations of the self organized neighbor embedding (SONE) for dimension reduction and visualization. In M. Verleysen, editor, *Proc. of 19th European Symposium on Artificial Neural Networks (ESANN)*, pages 29–34, Bruges, Belgium, April 2011.

[4] K. Bunte, M. Biehl, and B. Hammer. Supervised dimension reduction mappings. In M. Verleysen, editor, *Proc. of 19th European Symposium on Artificial Neural Networks (ESANN)*, pages 281–286, Bruges, Belgium, April 2011.

[5] K. Bunte, M. Biehl, and B. Hammer. Dimensionality reduction mappings. In *IEEE Symposium Series in Computational Intelligence (SSCI) 2011: Computational Intelligence and Data Mining (CIDM)*, pages 349–356, Paris, France, April 2011.

[6] M. B. Huber, K. Bunte, M. B. Nagarajan, M. Biehl, L. A. Ray, and A. Wismüller. Texture feature selection with relevance learning to classify interstitial lung disease patterns. In Ronald M. Summers M.D. and Bram van Ginneken, editors, *SPIE Medical Imaging: Computer-Aided Diagnosis*, volume 7963:43, March 2011.

[7] B. Hammer, K. Bunte, and M. Biehl. Some steps towards a general principle for dimensionality reduction mappings. In B. Hammer, P. Hitzler, W. Maas, and M. Toussaint, editors, *Learning paradigms in dynamic environments*, volume 10302 of *Dagstuhl Seminar Proceedings*, page 15. Schloss Dagstuhl, Leibniz Zentrum für Informatik, 2010.

[8] K. Bunte, B. Hammer, T. Villmann, M. Biehl, and A. Wismüller. Exploratory observation machine (XOM) with kullback-leibler divergence for dimensionality reduction and visualization. In M. Verleysen, editor, *Proc. of 18th European Symposium on Artificial Neural Networks (ESANN)*, pages 87–92, Bruges, Belgium, April 2010.

[9] K. Bunte, B. Hammer, and M. Biehl. Nonlinear dimension reduction and visualization of labeled data. In Xiaoyi Jiang and Nicolai Petkov, editors, *13th International Conference on Computer Analysis of Images and Patterns (CAIP)*, volume 5702 of *Lecture Notes in Computer Science*, pages 1162–1170, Münster, Germany, September 2009. Springer.

[10] K. Bunte, M. Biehl, N. Petkov, and M. F. Jonkman. Adaptive metrics for content based image retrieval in dermatology. In M. Verleysen, editor, *Proc. of 17th European Symposium on Artificial Neural Networks (ESANN)*, pages 129–134, Bruges, Belgium, April 2009.

[11] K. Bunte, B. Hammer, P. Schneider, and M. Biehl. Nonlinear discriminative data visualization. In M. Verleysen, editor, *Proc. of 17th European Symposium on Artificial Neural Networks (ESANN)*, pages 65–70, Bruges, Belgium, April 2009.

[12] T. Hermann, K. Bunte, and H. Ritter. Relevance-based interactive optimization of sonification. In *Lecture Presentation at the* 13^{th} *International Conference on Auditory Display (ICAD)*, pages 26–29, Montréal, Canada, June 2007.

Technical Reports

[1] Giuseppe Papari, Kerstin Bunte, and Michael Biehl. Waypoint averaging and step size control in learning by gradient descent. Technical Report MLR-2011-06, Leipzig University, 2011.

[2] K. Bunte, S. Haase, M. Biehl, and T. Villmann. Mathematical Foundations of Self Organized Neighbor Embedding (SONE) for Dimension Reduction and Visualization. Technical Report MLR-03-2010, Leipzig University, 2010.

[3] K. Bunte, P. Schneider, B. Hammer, F.-M. Schleif, T. Villmann, and M. Biehl. Discriminative visualization by limited rank matrix learning. Technical Report MLR-03-2008, Leipzig University, 2008.

[4] P. Schneider, K. Bunte, B. Hammer, T. Villmann, and M. Biehl. Regularization in matrix relevance learning. Technical Report MLR-02-2008, Leipzig University, 2008.

Submitted Papers

[1] Kerstin Bunte, Sven Haase, Michael Biehl, and Thomas Villmann. Stochastic neighbor embedding (SNE) for dimension reduction and visualization using arbitrary divergences. submitted to Neurocomputing, 2011.

Bibliography

Aeberhard, S., Coomans, D. and de Vel, O.: 1992, Comparison of classifiers in high dimensional settings, *Technical Report 02*, James Cook University.

Amari, S.-I.: 1985, *Differential-geometrical methods in statistics*, Springer, Berlin.

Amari, S. I. and Nagaoka, H.: 2000, Methods of information geometry, *Translations of Methematical Monographs*, Vol. 191, Oxford University Press, New York.

Asuncion, A., Newman, D. J., Hettich, S., Blake, C. L. and Merz, C. J.: 1998, UCI repository of machine learning databases, http://archive.ics.uci.edu/ml/.

Bae, S.-H., Choi, J. Y., Qiu, J. and Fox, G. C.: 2010, Dimension reduction and visualization of large high-dimensional data via interpolation, *Proc. of the 19th IEEE International Symposium on High Performance Distributed Computing (HPDC)*, HPDC '10, ACM, New York, NY, USA, pp. 203–214.

Banerjee, A., Dhillon, I., Ghosh, J., Merugu, S. and Modha, D. S.: 2004, A generalized maximum entropy approach to Bregman co-clustering and matrix approximation, *Proc. of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, New York, NY, USA, pp. 509–514.

Banerjee, A., Merugu, S., Dhillon, I. S. and Ghosh, J.: 2005, Clustering with Bregman divergences, *Journal of Machine Learning Research* **6**, 1705–1749.

Barbuddhe, S. B., Maier, T., Schwarz, G., Kostrzewa, M., Hof, H., Domann, E., Chakraborty, T. and Hain, T.: 2008, Rapid identification and typing of listeria species by matrix-assisted laser desorption ionization-time of flight mass spectrometry, *Applied and Environmental Microbiology* **74**(17), 5402–5407.

Bartlett, P. L. and Mendelson, S.: 2003, Rademacher and Gaussian complexities: risk bounds and structural results, *Journal of Machine Learning Research* **3**, 463–482.

Basu, A., Ian R. Harris, N. L. H. and Jones, M. C.: 1998, Robust and efficient estimation by minimising a density power divergence, *Biometrika* **85**(3), 549–559.

Baudat, G. and Anouar, F.: 2000, Generalized discriminant analysis using a kernel approach, *Neural Computation* **12**(10), 2385–2404.

Belkin, M. and Niyogi., P.: 2003, Laplacian eigenmaps for dimensionality reduction and data representation, *Neural Computation* **15**, 1373–15396.

Bensmail, H. and Celeux, G.: 1996, Regularized gaussian discriminant analysis through eigenvalue decomposition, *Journal of the American Statistical Association* **91**, 1743–1748.

Bertin, N., Fevotte, C. and Badeau, R.: 2009, A tempering approach for Itakura-Saito non-negative matrix factorization. with application to music transcription, *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE Computer Society, Washington, DC, USA, pp. 1545–1548.

Biehl, M., Ghosh, A. and Hammer, B.: 2007, Dynamics and generalization ability of LVQ algorithms, *Journal of Machine Learning Research* **8**, 323–360.

Bishop, C. M.: 1995, *Neural networks for pattern recognition*, Oxford University Press, USA.

Bishop, C. M.: 2006, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Bishop, C. M. and Williams, C. K. I.: 1998, GTM: The generative topographic mapping, *Neural Computation* **10**, 215–234.

Blum, A., Luedtke, H., Schwabe, U. E. R., Rassner, G. and Garbe, C.: 2004, Digital image analysis for diagnosis of cutaneous melanoma. Development of a highly effective computer algorithm based on analysis of 837 melanocytic lesions, *British Journal of Dermatology* **151**(5), 1029–1038.

Bojer, T., Hammer, B., Schunk, D. and von Toschanowitz, K. T.: 2001, Relevance determination in learning vector quantization, *in* M. Verleysen (ed.), *Proc. of the 9th European Symposium on Artificial Neural Networks (ESANN)*, Bruges, Belgium, pp. 271–276.

Bolognia, J. L., Jorizzo, J. L. and Rapini, R. P.: 2007, *Dermatology*, 2nd edn, Mosby.

Bosman, H. H. W. J., Petkov, N. and Jonkman, M. F.: 2010, Comparison of color representations for content based image retrieval in dermatology, *Skin Research and Technology*, Vol. 16, pp. 109–113.

Brand, M.: 2002, Charting a manifold, *Advances in Neural Information Processing Systems (NIPS)*, pp. 961–968.

Bregman, L. M.: 1967, The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming, *USSR Computational Mathematics and Mathematical Physics* **7**, 200–217.

Bunte, K., Biehl, M., Jonkman, M. F. and Petkov, N.: 2011, Learning effective color features for content based image retrieval in dermatology, *Pattern Recognition* **44**(9), 1892–1902.

Bunte, K., Giotis, I., Petkov, N. and Biehl, M.: 2011, Adaptive matrices for color texture classification, *14th International Conference on Computer Analysis of Images and Patterns (CAIP)*, Vol. 6855 of *Lecture Notes in Computer Science*, Seville, Spain, pp. 489–497.

Bunte, K., Hammer, B. and Biehl, M.: 2009, Nonlinear dimension reduction and visualization of labeled data, *in* X. Jiang and N. Petkov (eds), *13th International Conference on Computer Analysis of Images and Patterns (CAIP)*, Vol. 5702 of *Lecture Notes in Computer Science*, Springer, Münster, Germany, pp. 1162–1170.

Bunte, K., Hammer, B., Schneider, P. and Biehl, M.: 2009, Nonlinear discriminative data visualization, *in* M. Verleysen (ed.), *Proc. of 17th European Symposium on Artificial Neural Networks (ESANN)*, Bruges, Belgium, pp. 65–70.

Bunte, K., Hammer, B., Villmann, T., Biehl, M. and Wismüller, A.: 2010, Exploratory observation machine (XOM) with kullback-leibler divergence for dimensionality reduction and visualization, *in* M. Verleysen (ed.), *Proc. of 18th European Symposium on Artificial Neural Networks (ESANN)*, Bruges, Belgium, pp. 87–92.

Bunte, K., Hammer, B., Villmann, T., Biehl, M. and Wismüller, A.: 2011, Neighbor embedding XOM for dimension reduction and visualization, *Neurocomputing* **74**(9), 1340–1350.

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

Bunte, K., Schneider, P., Hammer, B., Schleif, F.-M., Villmann, T. and Biehl, M.: 2008, Discriminative visualization by limited rank matrix learning, *Technical Report MLR-03-2008*, Leipzig University.

Bunte, K., Schneider, P., Hammer, B., Schleif, F.-M., Villmann, T. and Biehl, M.: 2011, Limited rank matrix learning – discriminative dimension reduction and visualization, *Neural Networks* . Accepted for publication.

Carreira-Perpiñán, M. Á.: 2010, The elastic embedding algorithm for dimensionality reduction, *27th International Conference on Machine Learning (ICML) 2010*, pp. 167–174.

Chai, D. and Bouzerdoum, A.: 2000, A bayesian approach to skin color classification in YCbCr color space, *Proc. of the IEEE Region Ten Conference (TENCON)*, Vol. 2, pp. 421–424.

Cheng, Y., Swamisai, R., Umbaugh, S. E., Moss, R. H., W. V. Stoecker, W., Teegala, S. and Srinivasan, S. K.: 2008, Skin lesion classication using relative color features, *Skin Research and Technology*, Vol. 14, pp. 53–64.

Chopra, S., Hadsell, R. and Lecun, Y.: 2005, Learning a similarity metric discriminatively, with application to face verification, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Press, San Diego, CA, pp. 539–546.

Cichocki, A. and Amari, S.-I.: 2010, Families of Alpha- Beta- and Gamma-divergences: Flexible and robust measures of similarities, *Entropy* **13**, 134–170.

Cichocki, A., Zdunek, R., Phan, A. H. and Amari, S.-I.: 2009, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*, Hoboken, NJ: Wiley.

Collobert, R., Sinz, F., Weston, J. and Bottou, L.: 2006, Trading convexity for scalability, *Proc. of the 23rd International Conference on Machine Learning (ICML)*, ACM New York, NY, USA, pp. 201–208.

Cottrell, M., Hammer, B., Hasenfuss, A. and Villmann, T.: 2006, Batch and median neural gas, *Neural Networks* **19**(6-7), 762–771.

Cover, T. M. and Hart, P. E.: 1967, Nearest neighbor pattern classification, *IEEE Transactions on Information Theory* **13**(1), 21–27.

Crammer, K., Gilad-bachrach, R., Navot, A. and Tishby, N.: 2002, Margin analysis of the LVQ algorithm, *Advances in Neural Information Processing Systems (NIPS) 2002*, Vol. 15, MIT press, Cambridge, MA, USA, pp. 462–469.

Csiszár, I.: 1967, Information-type measures of difference of probability distributions and indirect observations, *Studia Scientarium Mathematicarum Hungarica*, Vol. 2, pp. 299–318.

Csiszár, I.: 1972, A class of measures of informativity of observation channels, *Periodica Mathematica Hungarica*, Vol. 2, pp. 191–213.

Datta, R., Li, J. and Wang, J. Z.: 2005, Content-based image retrieval: approaches and trends of the new age, *Proc. of the the 7th ACM International Workshop on Multimedia Information Retrieval*, ACM Press, pp. 253–262.

Dhillon, I. S. and Sra, S.: 2005, Generalized nonnegative matrix approximations with Bregman divergences, *Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada, pp. 283–290.

Dhillon, I. S. and Tropp, J. A.: 2007, Matrix nearness problems with Bregman divergences, *SIAM Journal on Matrix Analysis and Applications* **29**(4), 1120–1146.

Drimbarean, A. and Whelan, P. F.: 2001, Experiments in colour texture analysis, *Pattern Recognition Letters* **22**(10), 1161 – 1167.

Duda, R. O., Hart, P. E. and Stork, D. G.: 2000, *Pattern Classification*, Wiley-Interscience Publication.

Eguchi, S. and Kano, Y.: 2001, Robustifying maximum likelihood estimation, *Technical Report 802*, Tokyo-Institute of Statistical Mathematics, Tokyo.

Eisen, M. B., Spellman, P. T., Brown, P. O. and Botstein, D.: 1998, Cluster analysis and display of genome-wide expression patterns, *Proc. of the National Academy of Sciences (PNAS)* **95**(25), 14863–14868.

Ewing, R. M. and Cherry, J. M.: 2001, Visualization of expression clusters using sammon's non-linear mapping, *Bioinformatics* **17**, 658–659.

Faith, J., Mintram, R. and Angelova, M.: 2006, Targeted projection pursuit for visualising gene expression data classifications, *Bioinformatics* **22**, 2667–2673.

Felice, C. D., Flori, M. L., Pellegrino, M., Toti, P., Stanghellini, E., Molinu, A., Tosi, P. and Bagnoli, F.: 2002, Predictive value of skin color for illness severity in the high-risk newborn, *Pediatric Research* **51**(1), 100–105.

Fisher, R. A.: 1936, The use of multiple measurements in taxonomic problems, *Annals of Eugenics* **7**, 179–188.

Fogel, I. and Sagi, D.: 1989, Gabor filters as texture discriminator, *Biological Cybernetics* **61**(2), 103–113.

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

Friedman, J. H.: 1989, Regularized gaussian discriminant analysis, *Journal of the American Statistical Association* **84**, 165–175.

Frigyik, B. A., Srivastava, S. and Gupta, M.: 2008, An introduction to functional derivatives, *Technical Report UWEETR-2008-0001*, Seattle: Department of Electrical Engeneering, University of Washington.

Frome, A., Sha, F., Singer, Y. and Malik, J.: 2007, Learning globally-consistent local distance functions for shape-based image retrieval and classification, *Proc. of the 11th IEEE International Conference on Computer Vision (ICCV)*, Rio de Janeiro, Brazil.

Fujisawa, H. and Eguchi, S.: 2008, Robust parameter estimation with a small bias against heavy contamination, *Multivariate Analalysis* **99**(9), 2053–2081.

Fukunaga, K.: 1990, *Introduction to Statistical Pattern Recognition*, Computer Science and Scientific Computing Series, 2nd edn, Academic Press.

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

Ghosh, A., Biehl, M. and Hammer, B.: 2006, Performance analysis of lvq algorithms: A statistical physics approach, *Neural Networks* **19**(6-7), 817–829.

Giacinto, G. and Roli, F.: 2004, Bayesian relevance feedback for content-based image retrieval, *Pattern Recognition* **37**(7), 1499–1508.

Goldberger, J., Roweis, S., Hinton, G. and Salakhutdinov, R.: 2004, Neighborhood Component Analysis, *Advances in Neural Information Processing Systems (NIPS)*.

Graepel, T., Herbrich, R., Bollmann-Sdorra, P. and Obermayer, K.: 1999, Classification on pairwise proximity data, *Advances in Neural Information Processing Systems (NIPS) II*, MIT Press, Cambridge, MA, USA, pp. 438–444.

Grigorescu, S., Petkov, N. and Kruizinga, P.: 2002, Comparison of texture features based on Gabor filters, *IEEE Transactions on Image Processing* **11**(10), 1160–1167.

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

Haasdonk, B. and Bahlmann, C.: 2004, Learning with distance substitution kernels, *Pattern Recognition* **3175**, 220–227.

Hammer, B., Schleif, F.-M. and Villmann, T.: 2005, On the generalization ability of prototype basssed classifiers with local relevance determination, *Technical Report IfI-05-14*, Clausthal University of Technology.

Hammer, B., Strickert, M. and Villmann, T.: 2005a, On the generalization ability of GRLVQ networks, *Neural Processing Letters* **21**(2), 109–120.

Hammer, B., Strickert, M. and Villmann, T.: 2005b, Supervised neural gas with general similarity measure, *Neural Processing Letters* **21**(1), 21–44.

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

Han, J. and Kamber, M.: 2005, *Data Mining: Concepts and Techniques*, Morgen Kaufmann Publishers Inc.

Han, J. and Ma, K.-K.: 2007, Rotation-invariant and scale-invariant Gabor features for texture image retrieval, *Image and Vision Computing* **25**(9), 1474 – 1481.

Haralick, R. M., Shanmugam, K. and Dinstein, I.: 1973, Textural features for image classification, *IEEE Transactions on Systems, Man, and Cybernetics* **3**(6), 610 –621.

Hauta-Kasari, M., Parkkinen, J., Jaaskelainen, T. and Lenz, R.: 1999, Multi-spectral texture segmentation based on the spectral cooccurrence matrix, *Pattern Analysis and Applications* **2**, 275–284.

He, X., Cai, D., Yan, S. and Zhang, H.-J.: 2005, Neighborhood preserving embedding, *Proc. of the 10th IEEE International Conference on Computer Vision (ICCV)*, Vol. 2, pp. 1208 –1213.

He, X. and Niyogi, P.: 2003, Locality preserving projections, *Advances in Neural Information Processing Systems (NIPS) 16*, MIT Press.

Heskes, T.: 1999, Energy functions for Self-Organizing Maps, *in* S. Oja and E. Kaski (eds), *Kohonen Maps*, Elsevier, Amsterdam, pp. 303–316.

Hinton, G. and Roweis, S.: 2003, Stochastic neighbor embedding, *Advances in Neural Information Processing Systems (NIPS) 15*, MIT Press, pp. 833–840.

Hoang, M. A., Geusebroek, J.-M. and Smeulders, A. W.: 2005, Color texture measurement and segmentation, *Signal Processing* **85**(2), 265 – 275.

Hoffmann, K., Gambichler, T. and Rick, A.: 2003, Diagnostic and neural analysis of skin cancer (DANAOS). A multicentre study for collection and computer-aided analysis of data from pigmented skin lesions using digital dermoscopy, *British Journal of Dermatology* **149**(4), 801–809.

Itakura, F. and Saito, S.: 1968, Analysis synthesis telephony based upon the maximum likelihood method, *Independent Component Analysis*.

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

Jain, A. and Healey, G.: 1998, A multiscale representation including opponent color features for texture recognition, *IEEE Transactions on Image Processing* **7**(1), 124 –128.

Jain, A. K. and Farrokhnia, F.: 1991, Unsupervised texture segmentation using Gabor filters, *Pattern Recognition* **24**(12), 1167 – 1186.

Jain, A. K. and Vailaya, A.: 1996, Image retrieval using color and shape, *Pattern Recognition* **29**(8), 1233–1244.

Jau-Ling, S. and Ling-Hwei, C.: 2002, Color Image Retrieval Based on Primitives of Color Moments, *in* S.-K. Chang, Z. Chen and S.-Y. Lee (eds), *Recent Advances in Visual Information Systems*, Vol. 2314 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 19–27.

Jenssen, R.: 2005, *An Information Theoretic Approach to Machine Learning*, PhD thesis, University of Tromsø, Department of Physics.

Jenssen, R., Principe, J. C., Erdogmus, D. and Eltoft, T.: 2006, The Cauchy-Schwarz divergence and Parzen windowing: Connections to graph theory and Mercer kernels, *Journal of the Franklin Institute* **343**(6), 614–629.

Kakumanu, P., Makrogiannis, S. and Bourbakis, N.: 2007, A survey of skin-color modeling and detection methods, *Pattern Recognition* **40**, 1106–1122.

Kantorowitsch, I. and Akilow, G.: 1978, *Funktionalanalysis in normierten Räumen*, 2nd edn, Akademie-Verlag, Berlin.

Kapur, J. N.: 1994, *Measures of Information and their Applications*, Wiley-Interscience, Hoboken, NJ.

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

Khan, J., Wei, J. S., Ringnér, M., Saal, L. H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C. R., Peterson, C. and Meltzer, P. S.: 2001, Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks, *Nature Methods* **7**, 673–679.

Kietzmann, T. C., Lange, S. and Riedmiller, M.: 2008, Incremental GRLVQ: Learning relevant features for 3D object recognition, *Neurocomputing* **71**(13-15), 2868–2879.

Kjeldsen, R. and Kender, J.: 1996, Finding skin in color images, *Proc. of the 2nd International Conference on Automatic Face and Gesture Recognition*, Killington, Vermont, pp. 312–317.

Kohonen, T.: 1986, Learning vector quantization for pattern recognition, *Technical Report TKK-F-A601*, Helsinki University of Technology, Espoo, Finland.

Kohonen, T., Hynninen, J., Kangas, J., Laaksonen, J. and Torkkola, K.: 1996, LVQ PAK: The learning vector quantization program package, *Technical Report A30*, Helsinki University of Technology, FIN-02150 Espoo, Finland.

Kohonen, T. K.: 2002, *The handbook of brain theory and neural networks*, MIT press, Cambridge, MA, chapter Learning vector quantization, pp. 631–635.

Kohonen, T., Schroeder, M. R. and Huang, T. S. (eds): 2001, *Self-Organizing Maps*, 3rd edn, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Kontkanen, P., Lahtinen, J., Myllymäki, P., Silander, T. and Tirri, H.: 2000, Supervised model-based visualization of high-dimensional data, *Intelligent Data Analysis* **4**(3,4), 213–227.

Kruizinga, P. and Petkov, N.: 1995, A computational model of periodic-pattern-selective cells, *in* J. Mira and F. Sandoval (eds), *From Natural to Artificial Neural Computation*, Vol. 930 of *Lecture Notes in Computer Science*, Springer Berlin, Heidelberg, pp. 90–99.

Kulis, B., Surendran, A. and Platt, J.: 2007, Fast low-rank semidefinite programming for embedding and clustering, *Proc. of the 11th International Conference on Artificial Intelligence and Statistics (AI-STATS)*.

Kullback, S. and Leibler, R. A.: 1951, On information and sufficiency, *Annals of Mathematical Statistics* **22**, 49–86.

Lee, E. K., Cook, D., Klinke, S. and Lumley, T.: 2005, Projection pursuit for exploratory supervised classification, *Journal of Computational and Graphical Statistics* **14**(4), 831–846.

Lee, J. A., Archambeau, C. and Verleysen, M.: 2003, Locally linear embedding versus Isotop, *in* M. Verleysen (ed.), *Proc. of the 11th European Symposium on Artificial Neural Networks (ESANN)*, pp. 527–534.

Lee, J. A. and Verleysen, M.: 2005, Generalization of the l_p norm for time series and its application to self-organizing maps, *in* M. Cottrell (ed.), *Proc. of theWorkshop on Self-Organizing Maps (WSOM)*, Paris, Sorbonne, pp. 733–740.

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

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

Lee, J. and Verleysen, M.: 2007, *Nonlinear dimensionality reduction*, 1st edn, Springer.

Lehmann, T., Güld, M., Thies, C., Fischer, B., Spitzer, K., Keysers, D., Ney, H., Kohnen, M., Schubert, H. and Wein, B.: 2004, Content-based image retrieval in medical applications, *Methods of Information in Medicine*, Vol. 43(4), pp. 354–361.

Levina, E. and Bickel, P. J.: 2005, Maximum likelihood estimation of intrinsic dimension, *in* L. K. Saul, Y. Weiss and L. Bottou (eds), *Advances in Neural Information Processing Systems (NIPS)*, Vol. 17, MIT Press, Cambridge, USA, pp. 777 – 784.

Liese, F. and Vajda, I.: 1987, Convex statistical distances, *Teubner-Texte zur Mathematik*, Vol. 95, Teubner-Verlag, Leipzig.

Lyman, P. and Varian, H. R.: 2003, *How Much Information*. Retrieved from http://www.sims.berkeley.edu/how-much-info-2003 on October 31, 2011.

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

Mäenpää, T. and Pietikäinen, M.: 2004, Classification with color and texture: jointly or separately?, *Pattern Recognition* **37**(8), 1629 – 1640.

Maier, T., Klebel, S., Renner, U. and Kostrzewa, M.: 2006, Fast and reliable MALDI-TOF MS-based microorganism identification, *Nature Methods* (3).

Manjunath, B. and Ma, W.: 1996, Texture features for browsing and retrieval of image data, *IEEE Transactions Pattern Analysis and Machine Intelligence* **18**(8), 837 –842.

Martinetz, T. and Schulten, K.: 1991, A "neural-gas" network learns topologies, *Artificial Neural Networks* **I**, 397–402.

Memisevic, R. and Hinton, G.: 2005, Multiple relational embedding, *in* L. K. Saul, Y. Weiss and L. Bottou (eds), *Advances in Neural Information Processing Systems (NIPS) 17*, MIT Press, Cambridge, MA, pp. 913–920.

Mendenhall, M. J. and Merényi, E.: 2006, Generalized relevance learning vector quantization for classification driven feature extraction from hyperspectral data, *Proc. of the (ASPRS) Annual Conference and Technology Exhibition*, p. 8.

Messer, K. and Kittler, J.: 1999, A region-based image database system using colour and texture, *Pattern Recognition Letters* **20**(11-13), 1323 – 1330.

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

Mihoko, M. and Eguchi, S.: 2002, Robust blind source separation by beta divergence, *Neural Computation* **14**(8), 1859–1886.

Min, R. and Cheng, H. D.: 2009, Effective image retrieval using dominant color descriptor and fuzzy support vector machine, *Pattern Recognition* **42**(1), 147–157.

Mitchell, T. M.: 1997, *Machine Learning*, McGraw-Hill Series in Computer Science, WCB/McGraw-Hill, Boston, MA.

Mokbel, B., Gisbrecht, A. and Hammer, B.: 2010, On the effect of clustering on quality assessment measures for dimensionality reduction, *in* B. Hammer, F. Sha, L. van der Maaten and A. Smola (eds), *Proc. of the Advances in Neural Information Processing Systems (NIPS) Workshop on Challenges of Data Visualization*.

Möller, R. and Hoffmann, H.: 2004, An extension of neural gas to local PCA, *Neurocomputing* **62**(305-326).

Müller, H., Michoux, N., Bandon, D. and Geissbuhler, A.: 2004, A review of content-based image retrieval systems in medical applications – clinical benefits and future directions, *International Journal of Medical Informatics*, Vol. 73, pp. 1–23.

Murata, N., Takenouchi, T. and Kanamori, T.: 2004, Information geometry of U-Boost and Bregman divergence, *Neural Computation* **16**, 1437–1481.

Mwebaze, E., Schneider, P., Schleif, F.-M., Aduwo, J., Quinn, J., Haase, S., Villmann, T. and Biehl, M.: 2011, Divergence based classification in learning vector quantization, *Neurocomputing* **74**(9), 1429–1435.

Nene, S. A., Nayar, S. K. and Murase, H.: 1996, Columbia Object Image Library (COIL-20), *Technical Report CUCS-005-96*, Columbia University,.

Nielsen, F. and Nock, R.: 2009, Sided and symmetrized Bregman centroids, *IEEE Transactions on Information Theory* **55**, 2882–2904.

Ojala, T., Pietikäinen, M. and Mäenpää, T.: 2002, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, *IEEE Transactions Pattern Analysis and Machine Intelligence* **24**, 971–987.

Österreicher, F.: 2002, Csiszár f-divergences – basic properties, *Technical report*, Institute of Mathematics, University of Salzburg, Austria.

Palm, C.: 2004, Color texture classification by integrative co-occurrence matrices, *Pattern Recognition* **37**(5), 965 – 976.

Paschos, G.: 2000, Fast color texture recognition using chromaticity moments, *Pattern Recognition Letters* **21**(9), 837 – 841.

Pass, G., Zabih, R. and Miller, J.: 1996, Comparing images using color coherence vectors, *Proc. of the ACM Multimedia 96*, Boston, MA, pp. 65–73.

Peltonen, J., Goldberger, J. and Kaski, S.: 2006, Fast discriminative component analysis for comparing examples, *Proc. of the Advances in Neural Information Processing Systems (NIPS) Workshop on Learning to Compare Examples*.

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

Persson, P.-O. and Strang, G.: 2004, A simple mesh generator in matlab, *SIAM Review* **46**(2), 329–345.

Phung, S. L., Bouzerdoum, A. and Andchai, D.: 2002, A novel skin color model in YCbCr color space and its application to human face detection, *IEEE International Conference on Image Processing*, Vol. 1, pp. 289–292.

Pietikäinen, M., Ojala, T. and Xu, Z.: 2000, Rotation-invariant texture classification using feature distributions, *Pattern Recognition* **33**(1), 43 – 52.

Principe, J. C., Xu, D. and Fisher III, J. W.: 2000, Information-theoretic learning, *in* S. Haykin (ed.), *Unsupervised Adaptive Filtering*, second edn, Vol. 1, Wiley, New York, chapter 7.

Ramsay, J. and Silverman, B.: 2006, *Functional data analysis*, 2nd edn, Springer, New York.

Rényi, A.: 1960, On measures of entropy and information, *Proc. of the 4th Berkeley Symposium on Mathematics, Statistics and Probability*, Vol. 1, University of California Press, Berkeley, pp. 547–561.

Rényi, A.: 1970, *Probability Theory*, North-Holland series in applied mathematics and mechanics, v. 10, Amsterdam.

Ripley, B. D.: 1996, *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge.

Rogers, S. and Girolami, M.: 2007, Multi-class semisupervised learning with the e-truncated Multinomial Probit Gaussian Process, *Journal of Machine Learning Research*, Vol. 1 of *Gaussian Processes in Practice*, pp. 17 – 32.

Rossi, F., Delannay, N., Conan-Guez, B. and Verleysen, M.: 2005, Representation of functional data in neural networks, *Neurocomputing* **64**, 183–210.

Roweis, S. T. and Saul, L. K.: 2000, Nonlinear Dimensionality Reduction by Locally Linear Embedding, *Science* **290**(5500), 2323–2326.

Sammon, J. W.: 1969, A Nonlinear Mapping for Data Structure Analysis, *IEEE Transactions on Computers* **C-18**(5).

Sato, A. S. and Yamada, K.: 1996, Generalized learning vector quantization, *in* M. C. M. D. S. Touretzky and M. E. Hasselmo (eds), *Advances in Neural Information Processing Systems (NIPS)*, Vol. 8, MIT Press, Cambridge, MA, USA, pp. 423–429.

Sato, A. and Yamada, K.: 1998, An analysis of convergence in generalized LVQ, *in* L. Niklasson, M. Bodén and T. Ziemke (eds), *Proc. of the 8th International Conference on Artificial Neural Networks*, Vol. 1, Springer, London, pp. 170–176.

Scherf, U., Ross, D. T., Waltham, M. et al.: 2000, A gene expression database for the molecular pharmacology of cancer, *Nature Genetics* **24**, 236–244.

Schmid-Saugeona, P., Guillodb, J. and Thirana, J.-P.: 2003, Towards a computer-aided diagnosis system for pigmented skin lesions., *Computerized Medical Imaging and Graphics* **27**(1), 65–78.

Schneider, P., Biehl, M. and Hammer, B.: 2009a, Adaptive relevance matrices in learning vector quantization, *Neural Computation* **21**(12), 3532–3561.

Schneider, P., Biehl, M. and Hammer, B.: 2009b, Distance learning in discriminative vector quantization, *Neural Computation* **21**(10), 2942–2969.

Schneider, P., Bunte, K., Hammer, B. and Biehl, M.: 2010, Regularization in matrix relevance learning, *IEEE Transactions on Neural Networks* **21**(5), 831–840.

Schneider, P., Bunte, K., Hammer, B., Villmann, T. and Biehl, M.: 2008, Regularization in matrix relevance learning, *Technical Report MLR-02-2008*, Leipzig University.

Seo, S., Bode, M. and Obermayer, K.: 2003, Soft nearest prototype classification, *IEEE Transactions on Neural Networks* **14**, 390–398.

Seo, S. and Obermayer, K.: 2002, Soft learning vector quantization, *Neural Computation* **15**, 1589–1604.

Shannon, C. E.: 1948, *A Mathematical Theory of Communication*, CSLI Publications.

Shin, M. C., Chang, K. I. and Tsap, L. V.: 2002, Does colorspace transformation make any difference on skin detection?, *Proc. of the 6th IEEE Workshop on Applications of Computer Vision (WACV)*, IEEE Computer Society, Washington, DC, USA, pp. 275–279.

Skarbek, W. and Koschan, A.: 1994, Colour image segmentation – a survey, *Technical Report 32*, Technical University of Berlin, Department of Computer Science.

Smeulders, A., Worring, M., Santini, S., Gupta, A. and Jain, R.: 2000, Content-based image retrieval at the end of the early years, *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 22(12), pp. 1349–1380.

Sobottka, K. and Pitas, I.: 1996, Segmentation and tracking of faces in color images, *Proc. of the 2nd International Conference on Automatic Face and Gesture Recognition*, Killington, Vermont, pp. 236–241.

Song, L., Smola, A. J., Borgwardt, K. M. and Gretton, A.: 2008, Colored maximum variance unfolding, *in* J. C. Platt, D. Koller, Y. Singer and S. T. Roweis (eds), *Advances in Neural Information Processing Systems (NIPS)*, MIT Press, Cambridge, MA.

Sugiyama, M. and Roweis, S.: 2007, Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis, *Journal of Machine Learning Research* **8**, 1027–1061.

Sutton, R. S. and Barto, A. G.: 1998, Reinforcement learning: An introduction, *IEEE Transactions on Neural Networks* **9**(5), 1054–1054.

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

Takiwaki, H.: 1998, Measurement of skin color: Practical application and theoretical considerations, *Journal of Medical Investigation* **44**(3-4), 121–126.

Taneja, I. J. and Kumar, P.: 2004, Relative information of type s, Csiszár's f-divergence, and information inequalities, *Information Sciences-Informatics and Computer Science: An International Journal* **166**(1-4), 105–125.

Teh, Y. and Roweis, S.: 2003, Automatic alignment of local representations, *Advances in Neural Information Processing Systems (NIPS)* **15**, 841–848.

Tenenbaum, J. B., Silva, V. d. and Langford, J. C.: 2000, A global geometric framework for nonlinear dimensionality reduction, *Science* **290**(5500), 2319–2323.

Terrillon, J.-C. and Akamatsu, S.: 2000, Comparative performance of different chrominance spaces for color segmentation and detection of human faces in complex scene images, *Proc. of the 12th Conf. on Vision Interface (VI'99)*, Trois-Rivières, Canada, pp. 180–187.

Torgerson, W.: 1952, Multidimensional scaling, I: Theory and method, *Psychometrika* **17**, 401–419.

Torres, R. d. S., Falcão, A. X., Gonçalves, M. A., Papa, J. a. P., Zhang, B., Fan, W. and Fox, E. A.: 2009, A genetic programming framework for content-based image retrieval, *Pattern Recognition* **42**(2), 283–292.

Turner, M. R.: 1986, Texture discrimination by Gabor functions, *Biological Cybernetics* **55**, 71–82.

Umbaugh, S. E., Moss, R. H. and Stoecker, W. V.: 1992, An automatic color segmentation algorithm with application to identication of skin lesion borders, *Computerized Medical Imaging and Graphics* **16**, 227–235.

van der Maaten, L. and Hinton, G.: 2008, Visualizing data using t-SNE, *Journal of Machine Learning Research* **9**, 2579–2605.

van der Maaten, L. J. P.: 2009, Learning a parametric embedding by preserving local structure, *Proc. of the 12th International Conference on Artificial Intelligence and Statistics (AI-STATS)*, Vol. 5, JMLR W&CP, pp. 384–391.

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

Vandenberghe, L. and Boyd, S.: 1994, Semidefinite programming, *SIAM Review* **38**, 49–95.

Vasiloglou, N., Gray, A. and Anderson, D.: 2008, Scalable semidefinite manifold learning, *IEEE Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 368–373.

Venna, J.: 2007, *Dimensionality reduction for visual exploration of similarity structures*, PhD thesis, Helsinki University of Technology.

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

Vezhnevets, V., Sazonov, V. and Andreeva, A.: 2003, A survey on pixel-based skin color detection techniques, *Proc. of the 13th International Conference on Computer Graphics and Vision (Graphicon)*, pp. 85–92.

Villmann, T.: 2007, Sobolev metrics for learning of functional data – mathematical and theoretical aspects, *Technical Report MLR-03-2007*, Leipzig University.

Villmann, T. and Haase, S.: 2011, Divergence based vector quantization using Fréchet-derivatives, *Neural Computation* **23**(5), 1343–1392. accepted for publication.

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

Villmann, T., Merenyi, E. and Hammer, B.: 2003, Neural maps in remote sensing image analysis, *Neural Networks* **16**(3-4), 389–403.

Villmann, T. and Schleif, F.-M.: 2009, Functional vector quantization by neural maps, *in* J. Chanussot (ed.), *Proc. of the 1rst Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, IEEE Press, pp. 1–4.

VisTex: 2002, Vision Texture Database for color textures from MIT. available at: http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html.

Voigt, H. and Classen, R.: 2002, Computer vision and digital imaging technology in melanoma detection, *Seminars in Oncology* **29**(4), 308–327.

Wang, H., Zheng, J., an Yao, Z. and Li, L.: 2006, Improved locally linear embedding through new distance computing, *Advances in Neural Networks - ISNN*, Vol. 3971 of *Lecture Notes in Computer Science*, Chengdu, China, pp. 1326–1333.

Wang, J.-W., Chen, C.-H., Chien, W.-M. and Tsai, C.-M.: 1998, Texture classification using non-separable two-dimensional wavelets, *Pattern Recognition Letters* **19**, 1225–1234.

Wang, L. and Liu, J.: 1999, Texture classification using multiresolution markov random field models, *Pattern Recognition Letters* **20**(2), 171–182.

Weinberger, K. Q., Blitzer, J. and Saul, L. K.: 2006, Distance metric learning for large margin nearest neighbor classification, *Advances in Neural Information Processing Systems (NIPS)* **18**, 1473–1480.

Weinberger, K. Q. and Saul, L. K.: 2006, An introduction to nonlinear dimensionality reduction by maximum variance unfolding, *Proc. of the 21th National Conference on Artificial Intelligence* .

Wismüller, A.: 2001, Exploration-organized morphogenesis (XOM) – a general framework for learning by self-organization, *Human and Machine Perception*, Vol. 37 of *Reports of the Institute for Phonetics and Speech Communication (FIPKM)*, pp. 205–239.

Wismüller, A.: 2006, *Exploratory Morphogenesis (XOM): A Novel Computational Framework for Self-Organization*, PhD thesis, Technical University of Munich, Department of Electrical and Computer Engineering, Munich, Germany.

Wismüller, A.: 2009a, A computational framework for exploratory data analysis, *in* M. Verleysen (ed.), *European Symposium on Artificial Neural Networks (ESANN)*, Bruges, Belgium, pp. 547–552.

Wismüller, A.: 2009b, A computational framework for nonlinear dimensionality reduction and clustering, *in* J. Principe and R. Miikkulainen (eds), *Lecture Notes in Computer Science 5329*, Advances in Self-Organizing Maps, Springer, pp. 334–343.

Wismüller, A.: 2009c, The Exploration Machine – a novel method for structure-preserving dimensionality reduction, *in* M. Verleysen (ed.), *European Symposium on Artificial Neural Networks (ESANN)*, Bruges, Belgium, pp. 59–64.

Wismüller, A.: 2009d, The exploration machine: a novel method for analyzing high-dimensional data in computer-aided diagnosis, *Society of Photo-Optical Instrumentation Engineers (SPIE)*, Vol. 7260, pp. 72600G–72600G–7.

Wismüller, A.: 2011, Computational intelligence in biomedical imaging: Multidimensional analysis of spatio-temporal patterns, *Computer Science Research and Development* **26**(1), 15–37.

Xing, E. P., Ng, A. Y., Jordan, M. I. and Russell, S.: 2002, Distance metric learning, with application to clustering with side-information, *Advances in Neural Information Processing Systems (NIPS) 15*, MIT Press, Cambridge, pp. 505–512.

Zarit, B. D., Super, B. J. and Andquek, F. K. H.: 1999, Comparison of five color models in skin pixel classication, *International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems*, pp. 58–63.

Zhang, L., Zhang, A. and Ramanathan, M.: 2004, Vizstruct: exploratory visualization for gene expression profiling, *Bioinformatics* **20**, 85–92.

Zhang, Z. and Zha, H.: 2002, Principal manifolds and nonlinear dimension reduction via local tangent space alignment, *SIAM Journal of Scientific Computing* **26**, 313–338.

Zhao, L. and Zhang, Z.: 2009, Supervised locally linear embedding with probability-based distance for classification, *Computers & Mathematics with Applications* **57**(6), 919–926.

Samenvatting

Deze thesis presenteert een aantal extensies van het GLVQ algoritme gebaseerd op het concept van adaptive similarity measures. Deze metric learning kan worden gebruikt in een grote verscheidenheid aan applicaties, waaronder CBIR, supervised dimension reduction en advanced texture learning bij image analysis, om een paar te noemen. Het gedetailleerde onderzoek naar dimensionality reduction komt uitgebreid aan bod in de tweede helft van de thesis. Dit omvat onderzoek naar generalized explicit dimension reduction mappings voor unsupervised en supervised dimension reduction. Een nieuwe techniek voor efficient unsupervised non-linear dimension reduction wordt voorgesteld die de concepten van fast online learning en optimalisatie van divergenties combineert. Tot slot worden drie op divergentie gebaseerde algoritmes gegeneraliseerd en onderzocht op het gebruik van willekeurige divergenties.

In Chapter 2 wordt de benodigde achtergrond voor adaptive metric learning en prototype-based classification gegeven. Vervolgens wordt LiRaM LVQ geïntroduceerd in Chapter 3, een algoritme gericht op efficiënte optimalisatie van classificatie, met name bij zeer hoog-dimensionale datasets. Door de rank van de adaptieve matrix, een onderdeel van de gebruikte afstand, te begrenzen, kan het aantal vrije parameters expliciet worden gereguleerd. We laten zien dat naast computationele efficiëntie, het begrenzen van de rank een hogere kwaliteit laat zien vergeleken met alternatieve methoden gebaseerd op de decompositie van eigenwaarden na training, met name wanneer de target-dimensie lager is dan de intrinsieke dimensionaliteit van de dataset. Daarnaast staat dit concept discriminant linear dimension reduction toe, gericht op het behoud van de classification accuracy bij lagere dimensionaliteit. Door de distance measure in globale en lokale of klasse-specifieke matrices te ontbinden kunnen complexere decision boundaries worden bewerkstelligd in de visualisatie. Dit combineert linear dimension reduction met localized

similarity measures in laag-dimensionale ruimte, wat resulteert in non-linear decision boundaries van de receptieve velden. De dimension reduction met LiRaM LVQ toont vergelijkbare of betere resultaten dan alternatieve state-of-the-art technieken. Bovendien is de methode ook computationeel gezien efficiënt. In contrast met andere high-quality technieken vereist het niet de berekening van pair-wise affinities van de datapunten, maar slechts hun afstand tot het (kleine) aantal prototypes, wat over het algemeen minder berekeningen vereist. Verschillende experimenten op real-world datasets worden gepresenteerd en bevestigen onze claims.

Chapter 4 presenteert een voorbeeldapplicatie van LiRaM LVQ in de context van CBIR. Voor veel medische applicaties is de hoeveelheid data enorm gestegen in de afgelopen jaren. Daarom zijn computer aided diagnosis systems, die geautomatiseerd databases doorzoeken om potentieel interessante data voor een bepaalde taak voor te selecteren, zeer wenselijk. Dit werk behandelt CBIR in de context van dermatologie. In een samenwerkingsverband heeft de afdeling Dermatologie van het Universitair Medisch Centrum Groningen een database met afbeeldingen van verschillende typen huidletsels beschikbaar gesteld. Het doel is om gegeven een afbeelding een bepaald aantal vergelijkbare afbeeldingen op te leveren. Met het gebruik van adaptive metrics waren we in staat om het aandeel correct opgeleverde afbeeldingen aanzienlijk te verhogen, voor willekeurige color spaces. We vergelijken twee technieken voor distance learning: de LMNN en de LiRaM LVQ methode. Het is opmerkelijk dat LiRaM LVQ hierbij beter presteerde dan LMNN met typische instellingen. Door de complexiteit en het tijdsverbruik van LMNN te laten toenemen konden vergelijkbare resultaten worden behaald.

In Chapter 5 introduceren we een complexe variant op GLVQ voor texture classification, genaamd CIA LVQ. Deze flexibele methode combineert discriminative local linear projections in het Fourierdomein met linear filtering, e.g. met Gabor filters. Lineaire filteroperaties zijn vaak gedefinieerd op intensiteitswaarden. In het verleden zijn enkele heuristieke methoden voor filteroperaties op kleurenafbeeldingen voorgesteld die de response- of energiewaarden van kleurkanalen op een betekenisvolle manier combineren. Onze methode is van verschillende aard omdat het gebaseerd is op een automatisch lerende procedure gestuurd door supervised training. Hiervoor wordt a priori een Gabor filterbank verzameld met gewichten en oriëntaties passend bij de texture recognition taak. We nemen willekeurige segmenten van kleurenafbeeldingen van bekende klassen en voor elk van deze transformeren we de kleurkanalen afzonderlijk naar het Fourierdomein. De transformaties van kleurwaarden naar intensiteitswaarden worden geleerd door het CIA LVQ systeem om de filterresponses op deze getransformeerde segmenten beter te kunnen onderscheiden. In het bijzonder bij textures die zich in de natuur voordoen zoals schors en voedselstructuren presteert de voorgestelde techniek beter dan

alternatieve methoden waaronder het naïeve gebruik van een RGB naar grijswaarden transformatie, hetgeen in de praktijk vaak gebruikt wordt. Bovendien toont CIA LVQ uitstekende eigenschappen met betrekking tot evaluatie-afbeeldingen die niet eerder aan het systeem getoond zijn.

Deel II van deze thesis behandelt verschillende aspecten die betrekking hebben op dimension reduction. In Chapter 6 wordt een nieuwe algemene opvatting voorgesteld die de aanpassing van verschillende methoden voor dimension reduction voor explicit mappings vergemakkelijkt. In plaats van een impliciete optimalisatie van de posities van laag-dimensionale datapunten predefiniëren we de vorm van een mapping-functie f_W geparametriseerd door W, en optimaliseren we de parameters ten behoeve van een specifiek doel. Dit heeft het voordeel dat de training uitgevoerd kan worden op slechts een klein deel van de data en een rechtstreekse out-of-sample extensie voor alle datapunten is direct beschikbaar. Daarnaast wordt een theoretisch onderzoek naar de generalisatie-eigenschappen van dimension reduction mogelijk. We demonstreren het concept van dimension reduction mappings gebaseerd op de t-distributed SNE (t-SNE) kostenfunctie en verschillende alternatieven voor de mapping-functie f_W. Dit omvat zowel unsupervised linear en non-linear mappings gebaseerd op local PCA alsook supervised mappings die gebruikmaken van discriminative local linear projections. We vergelijken de methode met verschillende state-of-the-art technieken, tonen de uitstekende generalisatie-eigenschappen voor verschillende datasets en behandelen tenslotte het theoretische onderzoek naar dimension reduction mappings. In alle gevallen geeft onze methode vergelijkbare of zelfs betere resultaten.

Chapter 7 onderzoekt supervised dimension reduction gebaseerd op adaptieve afstanden en local linear projections verkregen door GMLVQ and LiRaM LVQ. Dit maakt de integratie van dimension reduction in de optimalisatieprocedure gericht op discriminative visualizations mogelijk. We laten zien met behulp van verschillende voorbeelden dat bestaande methoden voor dimension reduction uitgebreid kunnen worden naar een supervised setting gebruikmakend van de geleerde metrics en discriminative transformations van LVQ.

In Chapter 8 wordt een methode voor unsupervised dimension reduction voorgesteld, die fast sequential online learning combineert met direct divergence optimization zoals gebruikt in SNE en t-SNE. Deze techniek heet Self Organized Neighbor Embedding (SONE) en vertoont enkele interessante eigenschappen: in zijn oorspronkelijke formulering is SONE gebaseerd op een structuurhypothese die de gebruiker in staat stelt om het uiterlijk van de uiteindelijke embedding en de computationele inspanningen aan te passen. Veel technieken voor dimension reduction vereisen de berekening van alle pair-wise affinities van laag-dimensionale afbeeldingsvectoren in een optimalisatiestap. Dit heeft een computationele complexiteit

van $\mathcal{O}(n^2)$ tot gevolg, waarbij n staat voor het aantal datapunten. SONE berekent de afstanden naar één sampling vector uit de gegeven hypothese in iedere iteratie voor de aanpassing van alle punten. Daarmee is de computationele complexiteit lineair afhankelijk van het aantal punten en sampling vectors gegeven door de hypothese. Ondanks het feit dat de methode minder complex is dan SNE en t-SNE, toont het een vergelijkbare kwaliteit zoals gedemonstreerd wordt aan de hand van een aantal voorbeelden.

Chapter 9 behandelt een systematische aanpak voor de wiskundige behandeling van divergence based dimension reduction, zoals SNE, t-SNE en SONE, ten behoeve van de uitwisseling van hun respectievelijke modules. Naast de onafhankelijke behandeling van de verdeling in laag-dimensionale ruimte, e.g. het gebruik van een Gaussian voor SNE en een t-verdeling in t-SNE, concentreren we ons op de divergentie waarmee het verschil tussen verdelingen in de originele en de embeddingruimte gemeten wordt. Daarom bekijken we de divergentie-families en hun eigenschappen. We stellen een algemeen framework voor gebaseerd op het concept van Fréchet-afgeleiden en leiden de expliciete learning rules voor een breed scala aan divergenties af. In de experimenten concentreren we ons op de evaluatie van de Gamma-divergentie voor t-SNE en SONE in een aantal real-world datasets. We zagen dat de Gamma-divergentie de kwaliteit van de embeddings voor small neighborhoods verbetert vergeleken met de originele formulering met behulp van Kullback-Leibler.

Index