Hebbian learning approaches based on general inner products and distance measures in non-Euclidean spaces

Lange, Mandy

Hebbian Learning Approaches based on General Inner Products and Distance Measures in Non-Euclidean Spaces

Mandy Lange-Geisler

ISBN: 978-94-034-1470-6 (printed version)
ISBN: 978-94-034-1469-0 (electronic version)

university of groningen

Hebbian Learning Approaches based on General Inner Products and Distance Measures in Non-Euclidean Spaces

Phd thesis

to obtain the degree of PhD at the
University of Groningen
on the authority of the
Rector Magnificus prof. E. Sterken
and in accordance with
the decision by the College of Deans.

This thesis will be defended in public on

Monday 1 April 2019 at 11.00 hours

by

Mandy Lange-Geisler

born on 14 October 1984
in Altenburg, Duitsland

Supervisors
Prof. M. Biehl
Prof. T. Villmann

Assessment Committee
Prof. N. Petkov
Prof. B. Hammer
Prof. T. Martinetz

Contents

Acknowledgments

This thesis is finally completed. The last point has been made. With the support of some people I was able to realize this work in this form. Therefore I would like to take this opportunity to thank my PhD supervisors and colleagues for their scientific discussion, constructive criticism and general support, as well as I would like to thank you for the wonderful time during my PhD studies. Beside the scientific work, it was possible to attend several conferences in places of the world, which I probably never have be seen otherwise. Thus, the PhD time has become a very special stage in my life.

−Prof. Thomas Villmann−
thesis advisor: Thank you for support, guidance and for pushing me to go ahead. Many helpful friendly conversations have accompanied the PhD time. Thank you.

−Prof. Michael Biehl−
thesis advisor: In the final phase, he helped me find the right words. I would also like to thank you for your guidance assistance fighting the dutch bureaucracy.

−Marika Kaden−
diss-sis: proofreading and helpful discussions

−Tina Geweniger −
diss-sis: proofreading and useful hints

−David Nebel−
diss-bro: useful discussions

−Michiel Straat−
translator: the Dutch summary would not there without him

−Martin Sieber−
cover designer and preparing for printing

−my family −
Special thanks to my mother Angelika Lange and my husband Michael Geisler for believing in me and always supporting me when I needed. Especially, I want to thank my children for letting me take the time also to work on my thesis at home.

Abbreviations and Symbols

Abbreviations

ANN	**A**rtificial **N**eural **N**etworks
GD	**G**radient **D**escent
SGD	**S**tochastic **G**radient **D**escent
PCA	**P**rincipal **C**omponent **A**nalysis
MCA	**M**inor **C**omponent **A**nalysis
ICA	**I**ndependent **C**omponent **A**nalysis
KPCA	**K**ernel **PCA**
VQ	**V**ector **Q**uantization
LVQ	**L**earning **V**ector **Q**uantization
GLVQ	**G**eneralized **LVQ**
GMLVQ	**G**eneralized **M**atrix **LVQ**
LMQ	**L**earning **M**atrix **Q**uantization
GLMQ	**G**eneralized **LMQ**
GRMLVQ	**G**eneralized **R**elevance **LMQ**
HRL	**H**adamard-**R**elevance-**L**earning
MRL	**M**ultiplicative-**R**elevance-**L**earning
QR	**QR**-Relevance-Learning
KRL	**K**ronecker-**R**elevance-**L**earning
$\mathbf{GLMQ_{HRL}}$	**HRL** in **GLMQ**
$\mathbf{GLMQ_{l/r-MRL}}$	left/right **MRL** in **GLMQ**
$\mathbf{GLMQ_{QR}}$	**QR**-Relevance-Learning in **GLMQ**
$\mathbf{GLMQ_{KRL}}$	**K**ronecker-**R**elevance-**L**earning in **GLMQ**
SIP	**S**emi-**I**nner **P**roduct
gSIP	**g**eneralized **SIP**

gSIP(p)	**g**eneralized **SIP** of type **p**
GC-MS	**G**as **C**hromatography – **M**ass **S**pectrometry
TRLFS	**T**ime **R**esolved **L**aser induced **F**luorescence **S**pectroscopy
CLT	**C**entral **L**imit **T**heorem
pdfs	**p**robability **d**ensity **f**unction**s**
RKHS	**R**eproducing **K**ernel **H**ilbert **S**pace
RKBS	**R**eproducing **K**ernel **B**anach **S**pace
SIP-RKBS	**S**emi-**I**nner **P**roduct **RKBS**
TURL	**T**wo-**U**nit-**L**earning-**R**ule

Symbols

$\mathbf{x}$	vectors
$\mathbf{X}$	matrices
$\Delta\mathbf{x}$	update of $\mathbf{x}$
O	output of a neuron
Θ	threshold
η	learning rate
$\mathrm{idx}(\mathbf{x})$	identity function
$\mathrm{sign}(\mathbf{x})$	signum function
$H(x)$	Heaviside function
$\boldsymbol{\mu}$	mean vector
$\mathbf{C}$	data covariance matrix
λ_k	eigenvalues
$\mathbf{q}_k$	eigenvectors
$\mathbf{Q}$	PCA projections matrix
$\mathbf{Q}^-$	pseudo-inverse of $\mathbf{Q}$
$\mathrm{E}[\cdot,\cdot]$	expectation value
$E(\mathbf{x})$	cost function
$\nabla E(\mathbf{x})$	gradient of the cost function
$\mathbf{A}$	unknown mixing matrix
$\widetilde{\mathbf{A}}$	orthogonal mixing matrix
$\widetilde{\mathbf{a}}_i$	ICA basis vectors
$\mathbf{s}$	independent vectors

$\mathbf{E}$	orthogonal matrix consisting of eigenvectors of $\mathbf{C}$
$\mathbf{D}$	diagonal matrix containing the eigenvalues of $\mathbf{C}$
$\mathsf{kurt}\,(x)$	kurtosis
g	nonlinear function
$\widehat{m^4}$	estimation of the fourth moment
$\mathbf{w}$	prototype in LVQ
$\mathbf{w}^*$	winning prototype
$y\,(\mathbf{w})$	labeling of a prototypes $\mathbf{w}$
$\mathbf{v}$	data point
$x\,(\mathbf{v})$	labeling of a data point $\mathbf{v}$
$\hat{x}\,(\mathbf{v})$	predicted label of data point $\mathbf{v}$
$R_i\,(\mathbf{w}_j)$	receptive field R_i of prototype $\mathbf{w}_j$
err	classification error
acc	accuracy
$\Phi_{\cdot,\cdot}$	Kronecker delta function
$f_\Theta\,(x)$	sigmoid function
$d\,(\cdot,\cdot)$	dissimilarity measure
$d_E\,(\cdot,\cdot)$	squared Euclidean distance
$\|\mathbf{x}\|$	vector norm
$\|\mathbf{X}\|$	matrix norm
$\|\cdot\|_{l_p}$	l_p-norm
$\hat{\|\cdot\|}$	quasi-norm
$\langle\cdot,\cdot\rangle$	inner product
$[\cdot,\cdot]$	SIP
$\Re\,([\cdot,\cdot])$	real part of a SIP
$\mathcal{H}$	Hilbert space
$\mathcal{H}^n$	n-dimensional Hilbert space
$O_{\mathcal{H}^n}$	inner product of $\mathcal{H}^n$
$\mathbf{C}_{\mathcal{H}^n}$	covariance operator (matrix) in $\mathcal{H}^n$
$\mathscr{F}$	linear operator in $\mathcal{H}^n$
$\mathtt{H}$	countable basis
$\Omega_{\mathcal{H}}$	linear operator on $\mathcal{H}$
$\mathcal{B}$	Banach space
$\mathcal{B}^n$	n-dimensional Banach space
$O_{\mathcal{B}^n}$	semi-inner product of $\mathcal{B}$
$\mathbf{C}_{\mathcal{B}^n}$	covariance operator (matrix) in $\mathcal{B}^n$
B	finite basis in $\mathcal{B}^n$

$\mathcal{B}^*$	dual space of linear functionals over $\mathcal{B}$
B_s	Schauder basis
$\mathcal{B}_n \subset \mathcal{B}$	subspace of $\mathcal{B}$
$\mathcal{L}_p$	Lebesgue-integratable function
$\mathcal{W}_{K;p}$	Sobolev-space
D^α	differential operator of order $\lvert\alpha\rvert$
$\mathcal{K} \in \mathbb{R}^n$	compact set
κ_Φ	kernel function
$\Phi(\mathbf{v})$	feature map of $\mathbf{v}$
$\mathbf{C}_\Phi$	covariance matrix using Φ
$\mathbf{G}_m$	gram matrix containing inner products
$\mathbf{K}_m$	gram matrix containing semi-inner products
$\mathcal{I}_{\kappa_\Phi}$	image of V with κ_Φ
V_m	vector space with dimensionality m
$\mathscr{L}(V_m, V_n)$	vector space of linear functions between the vector spaces V_m and V_n
$\mathcal{B}_{m,n}$	Banach space of matrices
$\lVert\cdot\rVert_{\mathcal{S}p}$	Schatten-p-norm
$\operatorname{tr}(\cdot)$	trace operator
$\lvert\mathbf{A}\rvert$	absolute value of matrix $\mathbf{A}$
$\sigma(\mathbf{A})$	singular values of $\mathbf{A}$
$\mathbf{U}$	unitary matrix
$\mathbf{A}^*$	conjugate complex of $\mathbf{A}$
$\mathcal{S}_\alpha$	α-softmax function
$\mathcal{Q}_\alpha$	α-quasimax function
$\lvert x\rvert_\alpha^{\mathcal{S}}$	α-soft-absolute function
$\lvert x\rvert_\alpha^{Q}$	α-quasi-absolute function
$f_{em}(\omega)$	fluorescence intensity function
$f_{tr}(\tau)$	fluorescence light function

Chapter 1

Introduction

> *"When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place on one or both cells such that A's efficiency as one of the cells firing B, is increased."*
>
> Donald Hebb, 1949

A little more than a half a century ago D. Hebb proposed in "*The Organization of Behavior*" a hypothetical mechanism of cell assemblies, which are a group of nerve cells that can act like a form of short-term memory and sustainable reflect activity using the input itself. Hebb postulated a principle, how correlated features in stimuli should affect a change in neural connectivities such that coherent neural activity becomes more likely. This process characterizes biological neural learning and is the basis of artificial neural networks (ANN) and perceptron networks [111]. In ANNs the excitation of a neuron is determined by the Euclidean inner product between weights of neurons and the input vector. Later on, this so called *Hebbian learning* was adapted by neural network models such as neural maps [71] and others. These ANN approaches belong to the class of Hebbian like learning algorithms, in short Hebbian learning. Nowadays, many more machine learning algorithms belong to this class. Their neural network roots or their relation to them is no longer obvious at first glance.

Learning paradigms of machine learning algorithms can be mainly categorized into unsupervised, supervised, and reinforcement learning. Reinforcement learning is inspired by behaviorist psychology and is concerned with finding of suitable actions to

maximize some notion of reward [134]. Supervised methods involve external supervision, which provides correct responses to the given inputs. One objective of supervised learning is to learn the discrimination of classes and to maximize the generalization ability of the model. By contrast, unsupervised learning works without supervision and aims to discover hidden structures, regularities, features and correlations within the data [17]. For both, unsupervised and supervised learning, Hebbian approaches are known.

The machine learning methods, which follow the Hebbian principle, are widely applied in modern data analysis. For this purpose, highest neural excitation of ANNs corresponds to minimal Euclidean distance under normalization conditions for those models. However, data processing in non-Euclidean spaces is a currently challenging topic in machine learning data analysis [137]. For instance, it has been recognized that processing functional data with Sobolev distances is appropriate because the functional character of the data is taken into account [139].

The objective of this thesis is a unified and generalized scheme for Hebbian approaches in non-Euclidean spaces for unsupervised and supervised learning. This can be realized in different ways. One possibility is the replacement of the inner product by a semi-inner product (SIP). A SIP relaxes the strict properties of an inner product but preserves the linear aspect in the first argument. Thus, these SIPs are natural equivalents of inner products generating Banach spaces instead of Hilbert spaces for inner products. SIPs for Banach spaces are considered for applications with Hebbian like learning approaches. Famous examples of Banach spaces are l_p-spaces and Sobolev-spaces $\mathcal{W}_{K,p}$ for $p \neq 2$.

Since kernels correspond to inner products in a reproducing kernel Hilbert space (RKHS) the application of the kernel approach represents another possibility for Hebbian learning in non-Euclidean spaces. Here the data are implicitly mapped into a reproducing kernel Hilbert spaces (RKHS). Its inner product can be calculated from the original data using the kernel function. Thus, the kernel realizes an inner product in the Hilbert space and, hence, offers a new interpretation of Hebbian approaches that is based on it. In this thesis, the replacement of RKHS by reproducing kernel Banach space (RKBS) in Hebbian kernel methods, where the kernel is only a SIP, is considered [35, 88].

Most of the Hebbian learning schemes investigated in this work belong to unsupervised learning. However, the learning scheme of the supervised Learning Vector Quantization (LVQ) network, which is originally designed for applications in Euclidean data space, can be interpreted under specific circumstances as a Hebbian like learning, too.

Non-Euclidean metrics applied in LVQ can improve the performance of classification learning compared to standard approaches (Euclidean variants). Non-Euclidean LVQ variants can be obtained e. g. by means of l_p-norms, which represent another concern of this thesis.

The previously addressed Hebbian learning methods are vectorial approaches. However, if the data space is a vector space of matrices equipped with a respective matrix norm, then matrix approaches becomes of interest. Extensions of the Hebbian like learning methods in non-Euclidean spaces of matrices to process matrix data are the last main point of this thesis.

Outline

The following chapter 2 starts with Hebb's postulate of learning and its biological foundations in order to obtain a roughly mathematical model of the real biological model. This mathematical model forms the basis of all considered learning rules in this work. Its extension with a constrained Hebbian term yields the Oja algorithm, which determines iteratively the first principal component. Different variants and extensions of the Oja procedure are explained, i. e. several cost function based learning rules for PCA. Other simple Hebbian or anti-Hebbian learning rules, which can extract less dominant eigenvectors (minor components) or separate out independent components, are also which presented as a topic of chapter 2. Subsequently, LVQ along with several extended variants like Generalized LVQ (GLVQ) forms the last part of this fundamental chapter. Commonly all these methods are introduced in the Euclidean space.

Chapter 3 is based on the publications:

M. Lange and M. Biehl and T. Villmann, "Non-Euclidean Principal Component Analysis by Hebbian Learning", *Neurocomputing 147 (2015).[83]*

M. Biehl, M. Kästner, M. Lange and T. Villmann, "Non-Euclidean Principal Component Analysis and Oja's Learning Rule - Theoretical Aspects", *in P.A. Estevez and J.C. Principe and P. Zegers, ed., Advances in Self-Organizing Maps: 9th International Workshop WSOM 2012 Santiage de Chile vol. 198, (2013).[14]*

T. Villmann and M. Lange, "A comment on the functional L_p^{TS}-Measure Regarding the norm properties", *TechReport, 2015.[140]*

K. Domaschke, M. Kaden, M. Lange, T. Villmann, "Learning Matrix Quantization and Variants of Relevance Learning", *in M. Verleysen, ed., Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (2015). [32]*

A. Bohnsack, K. Domaschke, M. Kaden, M. Lange and T. Villmann, "Learning Matrix Quantization and Relevance Learning Based on Schatten-p-norms", *Neurocomputing 192 (2016).[19]*

A. Bohnsack, K. Domaschke, M. Kaden, M. Lange and T. Villmann, "Mathematical Characterization of Sophisticated Variants for Relevance Learning in Learning Matrix Quantization Based on Schatten-p-norms", *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science) 1 (2015).[18]*

A. Villmann, M. Lange-Geisler, T. Villmann, "About Semi-Inner Products for $\mathbf{p} - \mathbf{QR}$-Matrix Norms", *TechReport (2018).[136]*

In order to generate a uniform and general scheme for the Hebbian approaches in non-Euclidean spaces chapter 3 introduces the mathematical fundamentals of semi-inner products (SIPs) in Banach spaces and generalized SIPs, whose most important characteristics are presented in this context and conclusions are drawn. Known examples like the l_p-spaces, the Sobolev spaces and general kernel spaces equipped with their SIPs are considered more closely. For the Sobolev space, which is related to the l_p-space, a SIP is defined.

To create a matrix variant of the (vectorial) Hebbian approaches the last part of chapter 3 deals with vector spaces of matrices, i. e. matrix norms and their (semi)-inner products are addressed. Therefore, Schatten-p-norms are introduced and a respective SIP is defined. Further the **QR**-norm, which can be seen as generalization of Schatten-p-norms, is considered more closely.

The next chapters 4 and 5 comprise unsupervised and supervised Hebbian learning algorithms in non Euclidean spaces. Numerical simulations and selected applications are always given at the end of the chapters.

Chapter 4 is based on the publications:

M. Lange, M. Biehl, T. Villmann, "Non-Euclidean Principal Component Analysis by Hebbian Learning", *Neurocomputing, 2015.[83]*

M. Biehl, M. Kästner, M. Lange, T. Villmann, "Non-Euclidean Principal Component Analysis and Oja's Learning Rule - Theoretical Aspects", *in P.A. Estevez, J.C. Principe, P. Zegers, ed., Advances in Self-Organizing Maps: 9th International Workshop WSOM 2012 Santiago de Chile vol. 198, 2013.[14]*

M. Lange, M. Biehl, T. Villmann, "Non-Euclidean Independent Component Analysis and Oja's Learning", *in M. Verleysen, ed., Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (2013) .[80]*

M. Lange, D. Nebel and T. Villmann, "Non-Euclidean Principal Component Analysis for Matrices by Hebbian Learning", *in L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh and J.M. Zurada, ed., Artificial Intelligence and Soft Computing - Proc. the International Conference ICAISC vol. 8467, (2014).[81]*

More detailed, in chapter 4 Hebbian PCA learning for general finite dimensional Hilbert-spaces, which are isomorphic to the Euclidean space, are introduced, i. e. Hebbian PCA Learning is defined by means of an inner product of the Hilbert space. Furthermore it is shown that for Banach spaces the Hebbian PCA learning can be carried out using the underlying SIP. Moreover, it is also possible to extend the Hebbian PCA approach to RKHS and RKBS. These theoretical considerations of non-Euclidean PCA can be transferred to ICA. The focus in this work is on nonlinear ICA in general Reproducing Kernel spaces. The last theoretical part of the chapter 4 provides a matrix approach for Hebbian PCA learning based on Schatten-p-norms in the respective Banach space of matrices.

Chapter 5 is based on the publications:

M. Lange, T. Villmann, "Derivatives of l_p-norms and their Approximations", *Machine Learning Reports* 7, *MLR-04-2013 (2013). [79]*

M. Lange, D. Zühlke, O. Holz, T. Villmann, "Applications of l_p-norms and their Smooth Approximations for Gradient Based Learning Vector Quantization", *in M. Verleysen, ed., Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (2014). [82]*

M. Kaden, M. Lange, D. Nebel, M. Riedel and T. Geweniger and T. Villmann, "Aspects in Classification Learning - Review of Recent Developments in Learning Vector Quantization", *Foundations of Computing and Decision Sciences 39 (2014).[64]*

A. Bohnsack, K. Domaschke, M. Kaden, M. Lange, T. Villmann, "Learning Matrix Quantization and Relevance Learning Based on Schatten-p-norms", *Neurocomputing 192 (2016).[19]*

K. Domaschke, M. Kaden, M. Lange, T. Villmann, "Learning Matrix Quantization and Variants of Relevance Learning", *in M. Verleysen, ed., Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (2015). [32]*

Chapter 5 includes non-Euclidean variants of GLVQ with l_p-norms and their derivatives. Due to the inherent absolute value function in l_p-norms smooth approximations are required. Two different smooth approximations of the maximum function and their derivatives are discussed and smooth approximations of the absolute value function based on the maximum function and their derivatives are investigated. The last main point of this chapter is to provide a matrix approach of LVQ using the Schatten-p-norm or the **QR**-norm. The use of matrix norms leads to a greater structural flexibility of relevance learning and results in some new methods. Finally, a brief summary, concluding remarks and an outline for future work of the presented research are given in chapter 6.

Chapter 2

Hebbian Learning based on the Euclidean Inner Product

One of the most popular unsupervised algorithm using Hebbs paradigm is the learning rule proposed by E. Oja in [96], which is realizing an iterative Principal Component Analysis (PCA). PCA generates a basis of a multi-dimensional feature space, which reproduce the variability observed in given data. It determines the linear projection of the given data with respect to the largest variance as well as orthogonal directions, called principal components, ordered by decreasing variance [52]. Conventional algebraic approaches to PCA in which the eigenvectors of the empirical covariance matrix are directly calculated, are sensitive to outliers.

Hebbian PCA learning offers a more robust alternative. Other Hebbian-like methods, including Anti-Hebbian learning, perform Minor Component Analysis (MCA) and Independent Component Analysis (ICA). Whereas MCA is just the counterpart of PCA, i. e. the linear projection of the given data with respect to the smallest variance as well as orthogonal directions ordered by increasing variance, ICA constitutes a method to extract statistically independent sources from a sequence of mixtures, i.e. in general it is a tool for linear demixing of signals to detect the underlying independent sources [63].

At the beginning of this introductory chapter biological foundations of the structure and the functionality of special nerve of the nervous system are presented in order to obtain an abstract mathematical model of the real biological system. After that, several Hebbian approaches for PCA, MCA and ICA in the Euclidean space are

introduced. The last section deals with learning vector quantization (LVQ) and some extensions.

2.1 From Biological System to Hebbian Learning

Nerve cells or *neurons* are the functional units of the nervous system. A neuron is roughly structured into *dendrites*, *cell soma*, *axon* and *synapse*, see Figure 2.1(a). The dendrites, realizing the input of the neuron, receive information (stimuli) through the fine outgrowths from the environment and neighboring neurons and feed them to the soma containing the *nucleus* where the information is processed. After processing, the generated output is send over the *axion hillock* trough the *axon* to the synapse, which is the contact point to the dendrites of other adjacent neurons. [106]

Neurons exist in a variety of shapes and sizes, but all share more or less the same structure. Depending on their shape in the cerebral cortex two main classes of neurons can be distinguished: *pyramidal cells* and *stellate cells*. According to a widespread opinion, the essential information processing takes place in the pyramidal cells, which are named after the triangular shaped soma. The main structural features of pyramidal cells are the already mentioned triangular shaped soma, a single axon, and many dendrites, depicted in Figure 2.1(c). Rosenblatt studied in [111] this *pyramidal cell* to propose a highly simplified model of the nervous system. [110]

The mathematical model of such a pyramidal cell is denoted as *(simple) perceptron* (see Figure 2.1(b)). The information reaching the dendrites of pyramidal cells is modeled mathematically as an input vector (stimulus) $\mathbf{v} \in \mathbb{R}^m$. The information processing of the nucleus is assumed to be the weighted sum of the inputs $\sum_i w_i v_i = \mathbf{w}^\top \mathbf{v}$, where w_i are called the weights reflecting the dendritic connection strengths. All weights are collected in the weight vector $\mathbf{w} \in \mathbb{R}^m$. The output $O(\mathbf{v})$ of the perceptron is the modulated cell soma response

$$O(\mathbf{v}) = f\left(\mathbf{w}^\top \mathbf{v} - \Theta\right) \tag{2.1}$$

where f is called the transfer or activation function, which is the mathematical model of the axion hillock. This model of a pyramidal cell uses the Heaviside function

$$H(x) = \begin{cases} 1, & x \geq 0 \\ 0, & else \end{cases} \tag{2.2}$$

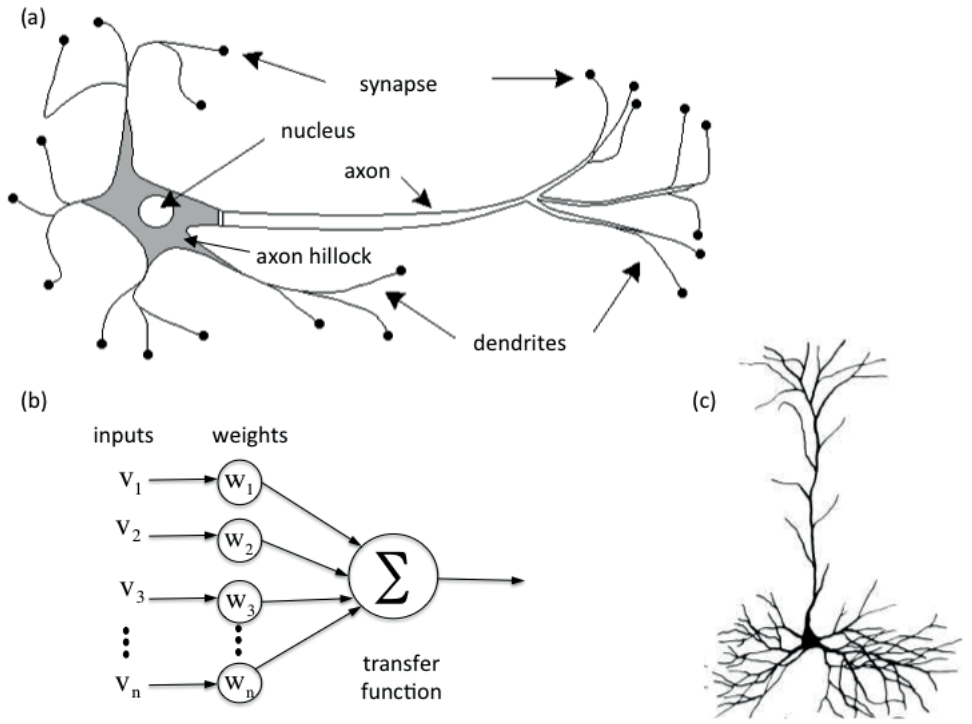

Figure 2.1: The illustration shows in Figure (a) the general schematic structure of a nerve cell. The mathematically model of a perceptron is depicted in Figure (b), which is derived from the biological model, the pyramidal cell to be seen in Figure (c).

as the activation function f in the cell soma, i. e. $f(x) = H(x)$ here. The threshold (bias) $\Theta \in \mathbb{R}$ models the activation level of the pyramidal cell and is responsible for the current state of a neuron. The output of a neuron causes an excitation in the neuron, if the sum $\mathbf{w}^\top \mathbf{v}$ exceeds the threshold Θ. The threshold Θ can also be formulated by an additional vector element w_{m+1} and constant input v_{n+1}. Therefore, in the description of neurons and their variants Θ will be omitted in future.

Depending on the transfer function several kinds of perceptrons can be distinguished. The linear perceptron, for instance, has a linear transfer function. In the following, a neuron corresponds always to a linear perceptron, i. e. $f(x) = x$, unless specified otherwise. The linear perceptron is pictured in Figure 2.2.

Learning in the nervous system takes place by adaptation of the dendritic connection strength $\mathbf{w}$ according to given stimuli. D. O. Hebb postulated in [47], that this

adaptation is proportional to the response $O(\mathbf{v})$, such that afterwards the cell is more adjusted to this stimulus. In the following this adaptation process is referred to *as Hebbian learning* and can be mathematically formulated by adapting the weight vector $\mathbf{w}$ for a given stimulus $\mathbf{v}$ in a perceptron

$$\begin{aligned}\mathbf{w}(t+1) &= \mathbf{w}(t) + \eta \cdot \Delta\mathbf{w} \\ &= \mathbf{w}(t) + \eta \cdot O \cdot \mathbf{v}(t),\end{aligned} \tag{2.3}$$

where $\eta \in \mathbb{R}$ with $0 < \eta \ll 1$ is called learning parameter. The excitation

$$O = \mathbf{w}^\top \mathbf{v} \tag{2.4}$$

of a neuron is merely the the Euclidean inner product $\langle \mathbf{w}, \mathbf{v} \rangle$. For a linear perceptron it is also the neuron output. Often, in this context the output O is referred to as *Hebb-output* or *Hebb-response*. Further, in the context of this work, the rule in (2.3) is also referred to as *Hebb rule*. The convergence of the Hebb rule to the global optima are secured by satisfying the condition $\sum_\iota \eta^2(\iota) < \infty$ (adiabatic decrease of the learning rates) and $\sum_\iota \eta(\iota) = \infty$ (infinite accumulated learning rate) [77], see Appendix A on page 131 for a more detailed explanation.

2.2 Hebbian Learning for Principal Component Analysis

Hebb's postulate of learning is the base of many online unsupervised learning algorithm like *Oja's learning rule* suggested by Oja (1982), which performs a Principal Component Analysis (PCA) and extracts only the first principal component. A generalization is *Sanger's learning rule,* which provides the possibility to determine all principal components. Further, these methods can be seen as a gradient descent of a cost function. However, both learning rules were originally proposed motivated heuristically. Modifications of Oja's learning rule and related cost functions are also introduced in this subsection, but first PCA is briefly introduced.

2.2.1 Principal Component Analysis (PCA)

Let $\mathbf{V} \subset \mathbb{R}^m$ be a data set of data vectors $\mathbf{v}_k \in \mathbb{R}^n$ with the mean vector $\boldsymbol{\mu} \in \mathbb{R}^m$. The data vectors $\mathbf{v}$ are orthogonally projected to

$$\tilde{\mathbf{v}} = \mathbf{Q}\,(\mathbf{v} - \boldsymbol{\mu}) \tag{2.5}$$

where $\mathbf{Q} \in \mathbb{R}^{m \times n}$ $(m \leq n)$ contains the PCA projection vectors $\mathbf{q}_k$ as its rows. These projection vectors are the eigenvectors of the sample covariance matrix $\mathbf{C} = \mathbf{V}^\top \mathbf{V}$ for centralized data, i. e. $\boldsymbol{\mu} = \mathbf{0}$. It is assumed, that $\mathbf{q}_k$ is sorted in a descending order according to the eigenvalues λ_k. The eigenvalues can be interpreted as the variance of the data along the eigen directions. The computation of the eigenvectors $\mathbf{q}_k$ takes place by solving the set of eigenvalue equations

$$\mathbf{C}\mathbf{q}_k = \lambda_k \mathbf{q}_k,\ k = 1, 2, ..., n. \tag{2.6}$$

Note that, $\mathbf{C}$ is a positive-semidefinite symmetric matrix with non-negative eigenvalues. The vectors $\mathbf{q}_k$ are known to be orthogonal. First, they are made orthonormal, i. e. the eigenvalues λ_k are proportional to the variance in the eigenvector directions. The proportion of variance retained by the PCA projection to k dimensions is described by the following normalized sum of these n eigenvalues:

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{j=1}^{n} \lambda_j} \geq \alpha \tag{2.7}$$

This condition indicates the number of the required dimensions to retain at least a proportion α of the variance in the PCA projection. Note that, two things of PCA:

- A PCA is the resulting uncorrelated representation $\tilde{\mathbf{v}}$ of the data vectors $\mathbf{v}$ and $\tilde{\mathbf{V}}$ the set of the linearly uncorrelated vectors called *principal components*.

- The PCA projection minimizes the squared reconstruction error

$$\sum_{\mathbf{v} \in \mathbf{V}} \left\| (\mathbf{v_i} - \boldsymbol{\mu}_i) - \mathbf{Q}^\top \mathbf{Q}\,(\mathbf{v_i} - \boldsymbol{\mu}_i) \right\|_{l_2}^2,\ \mathbf{Q} \in \mathbb{R}^{n \times n}. \tag{2.8}$$

  The second statement becomes evident, if the projected vector is projected back into the original space with $\mathbf{Q}^- \mathbf{Q}\,(\mathbf{v} - \mu)$, where $\mathbf{Q}^-$ denotes the pseudo-inverse of $\mathbf{Q}$. It is $\mathbf{Q}^- = \mathbf{Q}^\top$, since $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}$. Hence the squared reconstruction error in (2.8) is minimal and a PCA projection is an

optimal linear projection in the least squares reconstruction sense [48, 52].

Finding principal components reduces to finding the eigenvalues and eigenvectors of the covariance matrix $\mathbf{C}$. The eigenvalues are the roots of the characteristic polynomial of a square matrix. The algebraic calculation of the eigenvalues and eigenvectors of $\mathbf{C}$ by means of the characteristic polynomial is only possible for small matrices. Precisely, the eigenvalues and eigenvectors of an $n \times n$ matrix where $n > 4$ must be found numerically, because there are no analytical expressions for roots of polynomials with degree higher than 4.

One numerical algorithm for computing eigenvalues and eigenvectors was introduced in 1929, when Von Mises published the *power method* also called the *Von Mises iteration*. But this iterative method finds only the eigenvector corresponding to the largest absolute eigenvalue. Another method discovered by Jacobi in 1846 computes iteratively all eigenvalues und eigenvectors of real symmetric matrices and therefore also all principal components [76]. These iterative methods explicitly require the knowledge of the data covariance matrix $\mathbf{C}$ to determine the eigenvalues. In case of very high dimensional data the covariance matrix $\mathbf{C}$ becomes huge and the just mentioned iterative methods become inapplicable. As mentioned above, the learning algorithm as suggest by Oja (1982) offers an alternative to perform a PCA without the use of the covariance matrix $\mathbf{C}$ of given data. After convergence, $\mathbf{w}$ represents the first principal component [96].

2.2.2 Oja's Learning Rule for PCA

Let the inputs of the simple perceptron be n-dimensional column vectors $\mathbf{v} \in V$, which are centered as well as independently and identically distributed. In accordance with Hebb's postulate of learning, an adjustment of $\mathbf{w}$ takes place according to (2.3). This learning rule may lead to an unlimited growth of the synaptic weight vector $\mathbf{w}$ for example for constant inputs. This is unacceptable on biological grounds, because a synaptic connection cannot be of unlimited magnitude in the brain. This behavior can be avoided by constraining the growth of $\mathbf{w}$ by means of a normalization in the learning rule (2.3) as follows:

$$\mathbf{w}(t+1) = \frac{\mathbf{w}(t) + \eta O \mathbf{v}}{\|\mathbf{w}(t) + \eta O \mathbf{v}\|}. \tag{2.9}$$

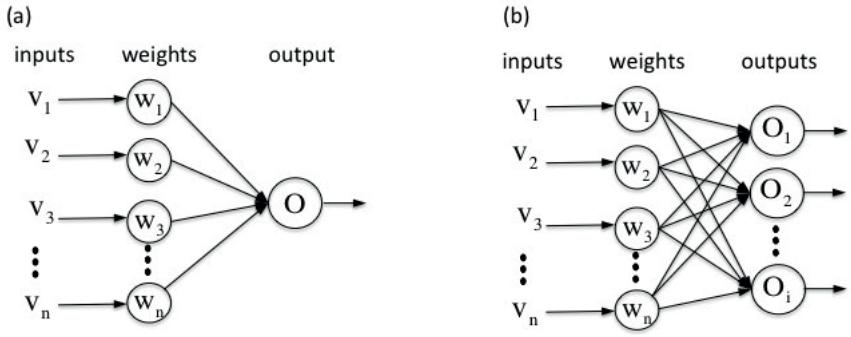

Figure 2.2: (a) structure of the perceptron model for Oja's algorithm, (b) extention for Sanger's algorithm.

A subsequent Taylor series expansion of (2.9) at $\eta = 0$ with the constraint $\|\mathbf{w}\| = 1$ yields the learning rule suggested by Oja in [96]

$$\begin{aligned}\mathbf{w}(t+1) &= \mathbf{w}(t) + \eta\left(O\mathbf{v}(t) - O^2\mathbf{w}(t)\right) \\ &= \mathbf{w}(t) + \eta\left(\mathbf{v}(t)\,\mathbf{v}^\top(t)\,\mathbf{w}(t) - \left(\mathbf{w}^\top(t)\left(\mathbf{v}(t)\,\mathbf{v}^\top(t)\right)\mathbf{w}(t)\right)\mathbf{w}(t)\right),\end{aligned} \tag{2.10}$$

known as *Oja's learning rule.* The term $O\mathbf{v}$ in (2.10) represents the usual Hebbian adaption step and $-O^2\mathbf{w}(t)$, resulted by normalization of (2.3), is responsible for stabilization. Further, the update scheme (2.10) represents a nonlinear difference equation, which makes it difficult to analyze convergence. The application of *Kushner's direct-averaging method* of this difference equation simplifies the convergence analysis [45]. This method assumes that $\mathbf{w}$ changes substantially slower in terms of magnitudes with respect to randomly selected inputs (data vectors) $\mathbf{v}$ by means of $0 < \eta \ll 1$. Hence, the averaged changes of $\mathbf{w}$ are considered instead of each step. Averaging the outer product $\mathbf{v}(t)\,\mathbf{v}^\top(t)$ yields the correlation matrix $\mathbf{C} = \mathrm{E}\left[\mathbf{v}\mathbf{v}^\top\right]$ defined by the expectation operator $\mathrm{E}[\cdot]$. Thus, Oja's averaged learning rule (2.10) becomes

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta\left(\mathbf{C}\mathbf{w}(t) - \langle\mathbf{w}(t), \mathbf{C}\mathbf{w}(t)\rangle\,\mathbf{w}(t)\right). \tag{2.11}$$

supposing slowly changing $\mathbf{w}$. The stationary state $\triangle\mathbf{w} = 0$ of averaged Oja's rule corresponds to the eigenvalue equation

$$\mathbf{C}\mathbf{w} = \langle\mathbf{w}, \mathbf{C}\mathbf{w}\rangle\,\mathbf{w}. \tag{2.12}$$

Moreover, the stability analysis by E. Oja in [96] shows that $\mathbf{w}$ converges in the stochastic update (2.10) to the eigenvector corresponding to the largest (absolute) eigenvalue of the correlation matrix $\mathbf{C}$. However, there are more than one fix point of the Oja algorithm, but they are not asymptotically stable and becomes the zero vector, i. e.$\mathbf{w} = \mathbf{0}$. [46].

As mentioned above, the learning scheme by Hebb and Oja can be seen as a gradient descent of a cost function. The existence of a cost function is a considerable advantage, because it simplifies the analysis of stable extrema by evaluating the Hessian matrix. Sompolinsky discovered that Hebb's rule in (2.3) is related to the cost function

$$E(\mathbf{w}) = -\frac{1}{2}\mathbf{w}^\top \mathbf{C}\mathbf{w} \tag{2.13}$$

with the gradient

$$\nabla E(\mathbf{w}) = -\mathbf{C}\mathbf{w} \tag{2.14}$$

[124]. Hebb's approach (2.3) is conform with a dampened Newton's method with learning rate η as damping rate. The cost function of the averaged version of Oja's learning rule (2.11) was an open problem until 1995. Zhang & Leung proposed [148] an appropriate cost function

$$E(\mathbf{w}) = -\mathbf{w}^\top \mathbf{w} - \ln\left(\mathbf{w}^\top C\mathbf{w}\right).$$

The respective minima are the same as the averaged version of Oja's learning rule (2.11).

2.2.3 Generalized Hebbian Algorithm

As previously emphasized, the Oja algorithm extracts only the first principal component. An extension of the linear perceptron model with several output nodes O_i, suggested by Sanger (1989) in [115], provides the possibility to determine more than one principal component by means of a generalized form of Hebb's paradigm, see Figure (2.2). This model yields the eigenvectors of $\mathbf{C}$ with respect to the corresponding eigenvalues in decreasing order by the adaption rule

$$\triangle \mathbf{w}_i = \eta\left(\mathbf{w}_i^\top \mathbf{v}\right)\left(\mathbf{v} - \sum_{j=1}^{i}\left(\mathbf{w}_j^\top \mathbf{v}\right)\mathbf{w}_j\right), \tag{2.15}$$

which here is called *Sanger's learning rule.* Note that, by using only one output node, i.e. $i = 1$, Sanger's learning rule (2.15) is simplified to Oja's learning rule. The stable fix points of Sanger's algorithm are all eigenvectors of the covariance matrix $\mathbf{C}$. A corresponding cost function of Sanger's learning rule is not known so far.

2.2.4 Further Learning Rules

Alternative learning rules for PCA are proposed by YUILLE in [145] and HASSOUN in [44], which are modifications of Oja's learning rule. A Hebbian-type adaption rule for $\mathbf{w}$ minimizes a cost function and results also the first principal component. Both learning rules are briefly stated in following.

YUILLE defines the cost function

$$E(\mathbf{w}) = -\frac{1}{2}\mathbf{w}^\top C\mathbf{w} + \frac{1}{4}\|\mathbf{w}\|^4 \tag{2.16}$$

to include inhibitory connections between neighboring units in the same layer realized by the second term $\frac{1}{4}\|\mathbf{w}\|^4$ [145]. The gradient of this cost function yields the *Yuille learning rule*

$$\begin{aligned}\mathbf{w}(t+1) &= \mathbf{w}(t) + \eta\left(C\mathbf{w}(t) - \|\mathbf{w}(t)\|^2\mathbf{w}(t)\right)\\ &= \mathbf{w}(t) + \eta\left(\mathbf{w}(t)^\top\mathbf{v}\mathbf{v} - \|\mathbf{w}(t)\|^2\mathbf{w}(t)\right).\end{aligned} \tag{2.17}$$

Let λ_{max} be the maximal absolute eigenvalue of $\mathbf{C}$. The weight vector $\mathbf{w}$ is constrained to $\|\mathbf{w}\| = \sqrt{\lambda_{max}}$ by the term $\|\mathbf{w}(t)\|^2\mathbf{w}(t)$ in learning rule (2.17). The extrema of the cost function (2.16) are either eigenvectors or a zero vector. Therefore, in (2.17) the weight vector $\mathbf{w}$ converges to the same maximal eigenvector direction as the learning rule by OJA. [48]

Another algorithm proposed by HASSOUN in [44] will be derived as gradient descent algorithm minimizing the following (Lagrangian) cost function

$$E(\mathbf{w}) = -\frac{1}{2}\mathbf{w}^\top C\mathbf{w} + \frac{\lambda}{2}\left(\|\mathbf{w}\| - 1\right)^2, \tag{2.18}$$

where $\lambda > 0$. These cost function incorporates the constraint $\|\mathbf{w}\| = 1$ to prevent the unlimited growth of $\mathbf{w}$ by the second term $(\|\mathbf{w}\| - 1)^2$. The minimization of (2.18) corresponds to a convex optimization problem. The gradient of cost function (2.18)

is referred to as *Hassoun's learning rule*

$$\begin{aligned}\mathbf{w}(t+1) &= \mathbf{w}(t) + \eta\left(C\mathbf{w}(t) - \lambda\left(1 - \frac{1}{\|\mathbf{w}(t)\|}\right)\mathbf{w}(t)\right) \\ &= \mathbf{w}(t) + \eta\left(\mathbf{w}(t)^{\top}\mathbf{v}\mathbf{v} - \lambda\left(1 - \frac{1}{\|\mathbf{w}(t)\|}\right)\mathbf{w}(t)\right). \end{aligned} \tag{2.19}$$

There are numerous variations of Oja's learning rule for different applications, such as Minor Component Analysis (MCA) and Independent Component Analysis (ICA), which are considered in the following sections.

2.3 Hebbian Learning for Minor Component Analysis

Further variants of Oja's learning rule can also perform other kinds of projection techniques such as Minor Component Analysis (MCA), which differs in only one point from PCA. Instead of principal components, the MCA extracts minor components. Minor components are defined as the eigenvectors corresponding to the smallest eigenvalues of the covariance matrix $\mathbf{C}$ of given data $\mathbf{V}$. Here, it is assumed that the eigenvectors are ordered according to an ascending variance. Hence, minor components are the counterparts of principal components. To solve the MCA problem, many neural learning algorithms have been proposed without calculating the covariance matrix in advance. In following an overview of the well-known algorithms for extracting the first minor component is given.

2.3.1 Oja's Learning Rule for MCA

Consider again a simple perceptron with input $\mathbf{v}$ and output $O = \mathbf{w}^{\top}\mathbf{v}$, where $\mathbf{w}$ is the weight vector $\mathbf{w}$. Oja proposed in [144] a learning algorithm, called Oja–Xu *algorithm*, to extract minor components from input data. The Oja–Xu algorithm is based on Oja's learning rule for PCA and results by changing the PCA learning rule into a constrained anti-Hebbian rule by reversing the sign and reads as follows:

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \eta\left(O\mathbf{v} - O^2\mathbf{w}(t)\right), \tag{2.20}$$

where η is a positive learning rate. However, this Oja MCA algorithm tends rapidly to infinite magnitudes of $\mathbf{w}$ and is not convergent. To guarantee the convergence, it

is necessary to use self-stabilizing algorithms. Thus, several variants of the OJA–XU *algorithm* are proposed. One modified variant is the OJAN *algorithm,* which includes a normalized anti-Hebbian rule:

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \eta O \cdot \left(\mathbf{v} - \frac{O\mathbf{w}(t)}{\mathbf{w}^{\top}(t)\,\mathbf{w}(t)} \right) \tag{2.21}$$

Often this leads to better convergence, but it may still be happen that $\mathbf{w}$ has infinite magnitudes. In [85], the stabilized version of Oja–Xu MCA learning algorithm, called OJA+ *algorithm*, is given by

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \eta \left(O\mathbf{v} - \left(O^2 + 1 - \|\mathbf{w}(t)\|^2 \right) \mathbf{w}(t) \right). \tag{2.22}$$

In order to guarantee the convergence, a further variant of the OJA–XU algorithm results by adding a normalization step

$$\mathbf{w}^*(t+1) = \frac{\mathbf{w}(t+1)}{\|\mathbf{w}(t+1)\|} \tag{2.23}$$

to the learning rule (2.20) and is called *modified* OJA–XU *MCA algorithm.* According to [101], the modified OJA–XU MCA algorithm has a bit higher computational complexity compared to OJA+ algorithm but a faster convergence speed.

2.4 Hebbian Learning for Independent Component Analysis

Independent Component Analysis (ICA) represents a technique to extract statistical independent sources from a sequence of mixtures [28]. A "source" means here an original data vector, i.e. independent component. Because of ICA can be seen as generalization of PCA a modified form of Oja's learning rule can also be applied here. This subsections starts with describing the problem of separating out statistical independent data vectors from noise and interferences. After that Oja's learning for ICA is stated.

2.4.1 Independent Component Analysis (ICA)

Let $\mathbf{s}(t) \in \mathbf{V} \subseteq \mathbb{R}^n$ be n-dimensional independent vectors, i. e. $s_i(t)$ and $s_j(t)$ are independent $\forall i \neq j$, which are mixed using an unknown mixing matrix $\mathbf{A}$ to get

n-dimensional mixture vectors $\mathbf{v}(t) \in V \subseteq \mathbb{R}^n$:

$$\mathbf{v}(t) = \mathbf{A}\mathbf{s}(t). \tag{2.24}$$

The goal of ICA is to estimate both the mixing matrix $\mathbf{A}$ and sources $\mathbf{s}(t)$, when only $\mathbf{v}(t)$ is known. Since both $\mathbf{s}$ and $\mathbf{A}$ are unknown the usual inverse of $\mathbf{A}$ cannot accomplish the purpose of the ICA. For reasons of simplicity, it is assumed that the number of independent components s_i is equal to the number of variables v_j. Hence the unknown mixing matrix $\mathbf{A}$ is square. Unlike sorted principal components obtained by PCA, the order of the independent sources remains unknown [133]. These sources are denoted as independent components.

As is well known, independence implies (nonlinear) uncorrelatedness, but not vice versa. Thus, independence is a much stronger property than uncorrelatedness, so that correlated solution possibilities can be rejected immediately. The decorrelation of $\mathbf{v}$ represents a reasonable preprocessing step after centralization of $\mathbf{v}$. Therefore, the majority of ICA algorithms requires a preliminary *sphering,* also referred to as *pre whitening.*

pre whitening A prominent approach for whitening, suggested by A. Hyvärien in [53], is based on eigenvalue decomposition of the data covariance matrix $\mathbf{C}$ with $\mathbf{C} = \mathbf{E}\mathbf{D}\mathbf{E}^\top$. Here, $\mathbf{E}$ is an orthogonal matrix consisting of eigenvectors of $\mathbf{C}$, which defines a rotation (change of coordinate axes) in $\mathbb{R}^n$ preserving norms and distances, and $\mathbf{D}$ is a diagonal matrix containing the respective eigenvalues of $\mathbf{C}$. The whitened variables $\widetilde{\mathbf{v}}$ results from

$$\widetilde{\mathbf{v}} = \mathbf{E}\mathbf{D}^{-\frac{1}{2}}\mathbf{E}^\top\mathbf{v}. \tag{2.25}$$

such that the expectation becomes $E\left(\widetilde{\mathbf{v}}\widetilde{\mathbf{v}}^\top\right) = \mathbf{I}$. This whitening process generates an orthogonal mixing matrix $\widetilde{\mathbf{A}}$ by

$$\begin{aligned} \widetilde{\mathbf{v}} &= \mathbf{E}\mathbf{D}^{-\frac{1}{2}}\mathbf{E}^\top\mathbf{A}\mathbf{s} \\ &= \widetilde{\mathbf{A}}\mathbf{s}. \end{aligned} \tag{2.26}$$

The column vectors of $\widetilde{\mathbf{A}}$ are denoted as $\widetilde{\mathbf{a}}_i$ and referred to as *ICA basis vectors.* Whitening can be performed by PCA and is frequently used. Therefore ICA is generally seen as an extension of PCA. Further, the previously performed whitening has the advantage that , firstly, the convergence of the ICA algorithm is speeded up considerably, secondly, noise may be decreased at the same time by the PCA sphering,

and thirdly, and the ICA algorithm will become somewhat stable [54].

After whitening, the goal remains to find a linear transformation for statistical independence of whitened vectors $\widetilde{\mathbf{v}}$. This may happen in a variety of ways. In general, ICA algorithms can be grouped into two broadly defined principles.

ICA estimation principle 1 (nonlinear decorrelation) *"Find the matrix $\widetilde{\mathbf{A}}$ so that for any $i \neq j$, the components s_i and s_j are uncorrelated, and the transformed components $\chi_1(s_i)$ and $\chi_2(s_j)$ are uncorrelated, where χ_1 and χ_2 are some suitable nonlinear functions."*[53]
Using this principle, ICA is performed by a stronger form of decorrelation. The nonlinearities χ_1 and χ_2 can be found by applying principles from estimation theory, such as the maximum-likelihood estimation [103], or from information theory via minimization mutual information by the Kullback-Leibler divergence [5] or the maximum-entropy-principle [10], where the last approach is known as Infomax algorithm.

ICA estimation principle 2 (maximum 'non-Gaussianity') *"Find the local maxima of 'non-Gaussianity' of a linear combination $s_i = \sum_{j=1}^{n} \widetilde{a}_{ij} \cdot \widetilde{v}_j$ under the constraint that the variance of s_i is constant. Each local maximum gives one independent component ."*[53].
This second principle maximizes the 'non-Gaussianity' using either the negentropy, which is based on the differential entropy [16], or kurtosis [54]. The Hebbian like learning algorithm for ICA, proposed by Hyvärinen & Oja in [54], uses implicitly the kurtosis.

2.4.2 Oja's Learning Rule for ICA

ICA estimation by Oja learning maximizes the 'non-Gaussianity', as just mentioned. The underlying idea is that according to the central limit theorem (CLT), sums of non gaussian random variables are closer to gaussian that the original ones [26, 56]. Precisely, let

$$s_i = \left\langle \widetilde{\mathbf{a}}_i^{\top}, \widetilde{\mathbf{v}} \right\rangle = \sum_{j=1}^{n} \widetilde{a}_{ij} \cdot \widetilde{v}_j \tag{2.27}$$

be the ith source. Here $\widetilde{v}_j$ are stochastic quantities such that the central limit theorem (CLT) is valid, i.e. the quantity s_i is more Gaussian than the individual summands. Thus, as indicated above, ICA can be performed by taking the absolute value of the

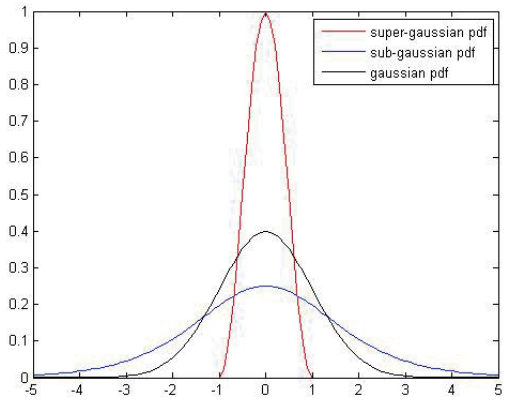

Figure 2.3: Probability density functions with mean 0, variance 1 and different kurtosis

kurtosis as a measure of the 'non-Gaussianity'. The kurtosis is defined as

$$\mathsf{kurt}\,(x) = E\left(x^4\right) - 3\left(E\left(x^2\right)\right)^2, \tag{2.28}$$

where the fourth moment $E\left(x^4\right)$ and the second moment $E\left(x^2\right)$ is used. The sign of the kurtosis depends on the *probability density functions* (pdfs) of $\mathbf{s}$. More precisely, the kurtosis of *super-Gaussian* pdfs is positive and for *sub-Gaussian pdfs* it is negative [**?**]. Super-Gaussian und sub-Gaussian pdfs are pictured in Figure 2.3. Frequently, there is no pre knowledge about the distribution of $\mathbf{s}$. Thus the ICA learning rule presented below uses as input whitened data and estimates the independent components without knowing whether the kurtosis has a positive or negative sign, i.e. the sign of the kurtosis is also estimated by introducing a second unit.

A General Two-Unit-Learning-Rule (TURL) for Whitened Data Hyvärinen & Oja proposed in [54] a learning rule based on a two unit system to separate out one source signal for whitened data. The vector $\mathbf{w} = \widetilde{\mathbf{a}}_i$ from (2.26) is interpreted again as a weight vector of a linear perceptron with the output $O(t) = \mathbf{w}(t)^\top \mathbf{v}(t)$, which is trained by a sequence of input vectors $\mathbf{v}(t)$ with the learning rate μ. In [54] was shown, that ICA can be performed by the learning rule

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \mu(t)\left[\sigma \mathbf{v}(t)\, g(O(t)) - \left(\|\mathbf{w}(t)\|^4 - 1\right)\mathbf{w}(t)\right], \tag{2.29}$$

where $g(x)$ is a nonlinear function in x such as the hyperbolic tangent $\tanh(x)$, which implicitly introduce the kurtosis. This can be seen by expanding $\tanh(x)$ into their Taylor series

$$\tanh(x) = x - \frac{1}{3}x^3 + \frac{2}{15}x^5 + \dots .$$

In general $g(x) = ax - bx^3$ with $a \geq 0$ and $b > 0$ is used in (2.29) as nonlinear function. Practically, any nonlinear function can be used for $g(x)$ to find independent components due to the derivation of (2.28) [53].
The term $\mathbf{v}(t)\, g(O(t))$ in (2.29) reflects the enhanced idea with the perceptron output learning function $g(O(t))$. Whereas the term $-\|\mathbf{w}\|^4 \mathbf{w}$ prevents $\mathbf{w}$ from infinite growing and $+\mathbf{w}$ prevents it from reaching the zero vector. For simplicity, it was specified that $g(x) = -x^3$ and thus the ICA learning rule reads as

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \mu(t) \left[\sigma \mathbf{v}(t) (O(t))^3 - \left(\left(\mathbf{w}(t)^\top \mathbf{w}(t) \right)^2 - 1 \right) \mathbf{w}(t) \right]. \quad (2.30)$$

The parameter $\sigma = \pm 1$ is a sign that determines whether the kurtosis is maximized ($\sigma = +1$) or minimized ($\sigma = -1$). The simultaneous determination of an appropriate σ requires a second unit $\widehat{m^4}(t)$, which estimates the kurtosis of the output $O(t)$ belonging to the first unit. Thus, σ is replaced by the sign of the estimated kurtosis. This yields a *general two unit learning rule for whitened data*

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \mu(t) \left[\operatorname{sign}\left(\widehat{\operatorname{kurt}}(t) \right) \mathbf{v}(t) (O(t))^3 - \left(\|\mathbf{w}(t)\|^4 - 1 \right) \mathbf{w}(t) \right], \quad (2.31)$$

with the estimated kurtosis

$$\begin{aligned} \widehat{\operatorname{kurt}}(t) &= \widehat{m^4}(t) - 3\|\mathbf{w}\|^4 \\ &= \widehat{m^4}(t) - 3\left(\mathbf{w}^\top \mathbf{w}\right)^2 \end{aligned} \quad (2.32)$$

using a separate estimation of the fourth moment by

$$\widehat{m^4}(t+1) = (1-\nu)\, \widehat{m^4}(t) + \nu (O(t))^4, \quad (2.33)$$

with $0 < \nu \ll 1$. After convergence, $\mathbf{w}$ represents one column of the mixing matrix. The general two unit learning rule (2.31) performs a stochastic gradient descent of the cost function

$$E(\mathbf{w}) = \sigma \left[\tfrac{1}{4} E\left\{ \left(\mathbf{w}^\top \mathbf{v} \right)^4 \right\} \right] - \tfrac{1}{3}\|\mathbf{w}\|^6 + \tfrac{1}{2}\|\mathbf{w}\|^2 . \quad (2.34)$$

The normalization of $\|\mathbf{w}\|$ in (2.34) takes place by $-\frac{1}{3}\|\mathbf{w}\|^6 + \frac{1}{2}\|\mathbf{w}\|^2$. Hyvärinen & Oja shows in [54] that $\mathbf{w}$ converge up to a constant to one of the columns of the transformed mixing matrix $\widetilde{\mathbf{A}}$ from (2.26).
There are numerous variations and extensions of the introduced ICA learning rule: Corresponding learning rules for non-sphered data can be obtained with a simple modification of the constraint term. Separating one independent component with positive (or negative) kurtosis represents just a special case of (2.31), where the second unit $\widehat{m^4}(t)$ is dropped. The estimation of all independent components can be determined by an extension of the learning rule (2.29) with several units. For further details see [54].

2.5 Hebbian Learning for Prototype based Supervised Vector Quantization

At the beginning of this chapter Hebb's postulate of learning was presented, which is now addressed again in connection with Learning Vector Quantization (LVQ). LVQ is one of the methods for supervised prototype based Vector Quantization (VQ). VQ can be distinguished between unsupervised and supervised approaches. Unsupervised VQ is an approved method for clustering and compressing very large datasets. The term 'prototype based' implies that a data set is represented by an essentially smaller number of prototypes. Some well-known methods are c-means [9], self-organizing maps (SOM) [73], and neural gas (NG) [89]. One characteristic common to all these methods is that a data point is uniquely assigned to its closest prototype in terms of the Euclidean distance.

Methods for supervised prototype based VQ deal generally with classification of labeled data, i. e. each data is assigned to a prototype. There exists a large variety of classification methods ranging from statistical models like Linear and Quadratic Discriminant Analysis (LDA/QDA) [114] to adaptive algorithms like the k-Nearest Neighbor (kNN) [30], Support Vector Machines (SVMs) [121], or LVQ [74], as indicated above. LVQ has the attractive feature of being very intuitive and plausible, in contrast to many other learning systems. The prototypes are defined in the same space as the input data and can be seen as typical representatives of their classes. This facilitates a straightforward interpretation of the classifier. In LVQ the similarity between prototypes and data points are calculated with an appropriate distance measure. A common choice is the Euclidean distance, as already mentioned. For normalized data, LVQ, along with its several variants, can also be interpreted as Hebbian-like-learning

scheme due to the relation between the Euclidean inner product $\mathbf{v}^\top \mathbf{w}$ and the squared Euclidean distance $d_E(\mathbf{v}, \mathbf{w}) = (\mathbf{v} - \mathbf{w})^2$, i. e.

$$\begin{aligned} \mathbf{v}^\top \mathbf{w} \text{ is maximized} &\Longleftrightarrow \mathbf{v}^\top \mathbf{v} - 2\mathbf{v}^\top \mathbf{w} + \mathbf{w}^\top \mathbf{w} \text{ is minimized} \\ &\Longleftrightarrow (\mathbf{v} - \mathbf{w})^\top (\mathbf{v} - \mathbf{w}) \text{ is minimized} \\ &\Longleftrightarrow |\mathbf{v} - \mathbf{w}| \text{ is minimized.} \end{aligned} \tag{2.35}$$

Thus, maximum excitation (see on page 10 eq. (2.4)) of a neuron corresponds to a distance minimization with LVQ, where now the weights $\mathbf{w}$ are referred to as prototypes. A more detailed description of LVQ and different variants is given after a short introduction of prototype based classification.

2.5.1 Prototype based Classification

Let $\mathbf{v}_t \in \mathbb{R}^n$, $t = 1, \ldots, m$ be data vectors of the input space $V \subset \mathbb{R}^n$ and $W = \{\mathbf{w}_k \in \mathbb{R}^n, k = 1, \ldots, l\}$ the set of prototypes $\mathbf{w}_k \in \mathbb{R}^n$, i. e. $\mathbf{w}_k$ are in the same space as the data vectors. Furthermore, for all prototypes exists a predefined class membership $y(\mathbf{w}) \in \mathcal{C}$, named *labeling*, such that each class is represented by at least one appropriately chosen prototype, assuming C classes.

The *nearest prototype classification* (NPC) is a very simple classifier, where an unlabeled data vector is assigned to the class of its nearest prototype. A nearest prototype classifier is parameterized by a set of labeled prototypes and a dissimilarity measure $d(\mathbf{v}, \mathbf{w})$, which is frequently the squared Euclidean distance. The classifier decision of the NPC performs a winner-takes-all decision by using

$$\mathbf{w}^* = \arg\min_k d(\mathbf{v}, \mathbf{w}_k). \tag{2.36}$$

This closest prototype $\mathbf{w}^*$ is also referred as *winning prototype* or *best matching prototype*. Its label $y(\mathbf{w}^*)$ determines the predicted class of the respective data vector $\mathbf{v}$. Thus, a tessellation of the input space, called *receptive fields,* is obtained

$$R_j = \{\mathbf{v} \in V \mid \mathbf{w}_j = \mathbf{w}^*\}.$$

Hence, exactly one prototype $\mathbf{w}_j$ belongs to a receptive field R_j representing a subset of the input space, see Figure (2.4). Classification by NPC takes place by an assignment only. Learning of prototypes can be performed for example with LVQ.

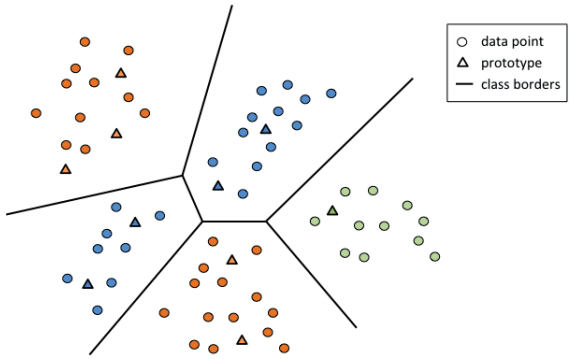

Figure 2.4: Class borders of a three-class problem, where each class is represented by a number of prototypes.

2.5.2 Basic Principles of LVQ and Variants

LVQ algorithms, introduced by T. Kohonen [71], are some of the most successful classifiers. There are numerous variants of LVQ with many extensions realizing different learning schemes. The basic approaches are the algorithms LVQ1 ... LVQ3 [74]. Recently developed extensions and modifications are explained in [64, 95].

LVQ algorithms require a set of prototypes $W = \{\mathbf{w}_k \in \mathbb{R}^n,\, k = 1, \ldots, l\}$ with class labels $y(\mathbf{w}) \in \mathcal{C}$ such that each class is represented by at least one prototype. The training data $\mathbf{v}_t \in \mathbb{R}^n,\, t = 1, \ldots, m$ of the input space $V \subset \mathbb{R}^n$ are labeled by $x(\mathbf{v}) \in \mathcal{C}$ such that each vector $\mathbf{v}$ belongs to a class. The task of all LVQ models is to find a model that assigns a data point to a predicted label $\hat{x}(\mathbf{v}) \in \mathcal{C}$ from the point of view of correctness, i.e. good classification performance. This can be measured by the *classification accuracy*

$$acc(V, W) = \frac{1}{m} \sum_{\mathbf{v} \epsilon V} \Phi_{x(\mathbf{v}), \hat{x}(\mathbf{v})} \tag{2.37}$$

or its equivalent the *classification error* $err(V, W) = 1 - acc(V, W)$, where $\Phi_{x(\mathbf{v}), \hat{x}(\mathbf{v})}$ with

$$\Phi_{x(\mathbf{v}), \hat{x}(\mathbf{v})} = \begin{cases} 1, & \text{if } x(\mathbf{v}) = \hat{x}(\mathbf{v}) \\ 0, & \text{else} \end{cases} \tag{2.38}$$

is the *Kronecker delta function.*

LVQ1

The first version of LVQ, introduced by KOHONEN, is a heuristic learning scheme designed to approximate a *Bayes* classification scheme in an intuitive way [71]. In each iteration of the learning process a randomly presented input vector $\mathbf{v} \in V$ causes an update of the best matching prototype $\mathbf{w}^*$. Depending on the class label evaluation, the prototype is moved towards to $\mathbf{v}$ by

$$\mathbf{w}^* \leftarrow \mathbf{w}^* + \eta_{\mathbf{w}} \cdot (\mathbf{v} - \mathbf{w}^*), \quad \text{if } x(\mathbf{v}) = y(\mathbf{w}^*) \tag{2.39}$$

if they belong to the same class or the prototype is pushed away from $\mathbf{v}$ by

$$\mathbf{w}^* \leftarrow \mathbf{w}^* - \eta_{\mathbf{w}} \cdot (\mathbf{v} - \mathbf{w}^*), \quad \text{if } x(\mathbf{v}) \neq y(\mathbf{w}^*), \tag{2.40}$$

in case of different classes, where $0 < \eta_{\mathbf{w}} \ll 1$ denotes a learning rate for the prototypes. These updates can be interpreted as Hebbian learning due to the relation in (2.35). Further, the update rules (2.39) and (2.40) can be written in a more general way taking into account that the squared Euclidean distance $d_E(\mathbf{v}, \mathbf{w}_j) = \|\mathbf{v} - \mathbf{w}_j\|_{l_2}^2$ is applied for winner determination:

$$\begin{aligned} \mathbf{w}^* &\leftarrow \mathbf{w}^* - \eta_{\mathbf{w}} \cdot \frac{1}{2} \cdot \frac{\partial d_E(\mathbf{v}, \mathbf{w}^*)}{\partial \mathbf{w}^*}, \quad \text{if } x(\mathbf{v}) = y(\mathbf{w}^*) \\ \mathbf{w}^* &\leftarrow \mathbf{w}^* + \eta_{\mathbf{w}} \cdot \frac{1}{2} \cdot \frac{\partial d_E(\mathbf{v}, \mathbf{w}^*)}{\partial \mathbf{w}^*}, \quad \text{if } x(\mathbf{v}) \neq y(\mathbf{w}^*), \end{aligned} \tag{2.41}$$

where $\frac{\partial d_E(\mathbf{v}, \mathbf{w}^*)}{\partial \mathbf{w}^*} = -2(\mathbf{v} - \mathbf{w}^*)$. After training a new data vector $\mathbf{v} \in \mathbb{R}^n$ is assigned to a class using winner takes all rule (2.36).
In LVQ1 only the closest prototype is updated at each step. Further modifications of LVQ1 were made by KOHONEN aiming at better convergence or favorable generalization behavior. In LVQ2.1 the two closest prototypes to $\mathbf{v}$ are updated simultaneously subject to some window-rule controlling the drift of the prototypes to avoid divergence. One of the two closest prototypes belongs to the correct class and the other to a wrong class, respectively. LVQ3 is identical to LVQ2.1, but include an additional learning rule to intercept that the two closest prototypes belong to the same class. In general terms, the original LVQ variants (LVQ1 ... LVQ3) differ in their particular training schemes, however, all realize after learning an approximated Bayes-classifier [72]. One major issue of these models is that the underlying learning rules are only heuristically motivated.

Generalized LVQ

The *Generalized Learning Vector Quantization* (GLVQ), proposed by Sato & Yamada in [116], is a modification of the intuitive LVQ algorithm and overcomes the problem of the non-existing cost function. The cost function can be perceived as a function that approximates the classification error with the objective to be minimized or as a function that approximates classification accuracy with the objective to be maximized. Howsoever, the advantage of a cost function based approach is that the optimization can be executed by gradient based methods and is no longer a heuristic. Sato & Yamada introduced the classifier function

$$\mu(\mathbf{v}) = \frac{d^+(\mathbf{v}) - d^-(\mathbf{v})}{d^+(\mathbf{v}) + d^-(\mathbf{v})} \in [-1, 1], \tag{2.42}$$

where $d^+(\mathbf{v}) = d(\mathbf{v}, \mathbf{w}^+)$ denotes the dissimilarity between the data vector $\mathbf{v}$ and the closest prototype $\mathbf{w}^+$ with coinciding class labels $y(\mathbf{w}^+) = x(\mathbf{v})$ and $d^-(\mathbf{v}) = d(\mathbf{v}, \mathbf{w}^-)$ is the dissimilarity value for the best matching prototype with a class label $y(\mathbf{w}^-)$ different from $x(\mathbf{v})$. Hence, $\mu(\mathbf{v}) < 0$ iff a data sample $\mathbf{v}$ is correctly classified. The classifier function $\mu(\mathbf{v})$ is in the range $[-1, 1]$ due to the normalization term $d^+(\mathbf{v}) + d^-(\mathbf{v})$ in (2.42).
The GLVQ cost function is defined by

$$E_{GLVQ} = \frac{1}{2 \cdot N_V} \sum_{\mathbf{v} \in V} f(\mu(\mathbf{v})), \tag{2.43}$$

where N_V denotes the cardinality of V and f is a monotonically increasing transfer or squashing function. Frequently, f is chosen as the identity function $f(x) = x$ or as differentiable sigmoid function

$$f_\Theta(x) = \frac{1}{1 + \exp\left(-\frac{x}{2\Theta^2}\right)}. \tag{2.44}$$

The parameter Θ in (2.44) refers to the slope, i.e. the smaller Θ the steeper the slope, see Figure (2.5).It can also be seen in this Figure that, the summands in (2.43) are in the range $[0, 1]$. Hence, the cost function with the sigmoid function is a smooth approximation of the classification error for $\Theta \searrow 0$ [65].
Learning in GLVQ is performed by stochastic gradient descent (SGD) for the cost function E_{GLVQ} (2.43). The SGD is explained in Appendix A. In each learning step of GLVQ, the winning prototypes $\mathbf{w}^+$ and $\mathbf{w}^-$ are adapted concurrently for a randomly chosen training datapoint $\mathbf{v} \in \mathbf{V}$, see Figure 2.6. The stochastic derivatives

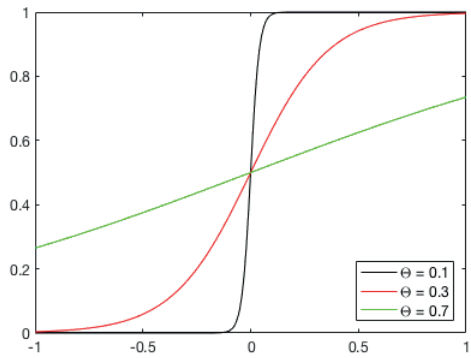

Figure 2.5: Representation of different shapes of the sigmoid function depending on Θ

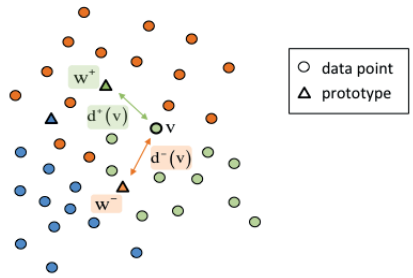

Figure 2.6: Nearest prototype determination $\mathbf{w}^+$ (coincide class labels) and $\mathbf{w}^-$ (different class labels) together with their distances $d^+(\mathbf{v})$ and $d^-(\mathbf{v})$, respectively. The data set realizes a three-class problem, where each class is represented by one prototype.

of E_{GLVQ} with respect to $\mathbf{w}^+$ and $\mathbf{w}^-$ yield the updates for the prototypes

$$\mathbf{w}^{\pm} \leftarrow \mathbf{w}^{\pm} - \eta_{\mathbf{w}} \cdot \triangle \mathbf{w}^{\pm}, \tag{2.45}$$

where

$$\triangle \mathbf{w}^{\pm} \sim \frac{\partial f\left(\mu\left(\mathbf{v}\right)\right)}{\partial \mathbf{w}^{\pm}} \tag{2.46}$$

$$= \frac{\partial f}{\partial \mu} \cdot \frac{\partial \mu}{\partial d_E^{\pm}\left(\mathbf{v}\right)} \cdot \frac{\partial d_E^{\pm}\left(\mathbf{v}\right)}{\partial \mathbf{w}^{\pm}} \tag{2.47}$$

$$= \frac{\partial f}{\partial \mu} \cdot \mp 2 \cdot \frac{d_E^{\mp}\left(\mathbf{v}\right)}{\left(d_E^{+}\left(\mathbf{v}\right) + d_E^{-}\left(\mathbf{v}\right)\right)^2} \cdot \frac{\partial d_E^{\pm}\left(\mathbf{v}\right)}{\partial \mathbf{w}^{\pm}}, \tag{2.48}$$

where squared Euclidean distance $d_E\left(\mathbf{v}, \mathbf{w}\right)$ is applied as dissimilarity measure. Instead of Euclidean distance any dissimilarity measures differentiable with respect to the prototypes can be applied such as l_p-norms or kernels, see chapter 5. Up to now, only the adaptation of the prototypes has been addressed. However, the possibility of distance adaptation, i.e. additional learning of the distance parameters by SGD, can improve the classification performance, which is realized by the following procedures.

Relevance Learning in Generalized LVQ

A successful extension of GLVQ is the *Generalized Relevance Learning Vector Quantization* (GRLVQ) proposed by Hammer & Villmann in [40]. The idea of relevance learning is that all data dimension are weighted according to their relevance for a better classification performance of GLVQ. Thus, the extended variant inherits the same cost function (2.43) replacing the squared Euclidean distance by the weighted variant

$$d_{E,\boldsymbol{\lambda}}\left(\mathbf{v}, \mathbf{w}\right) = \left(\boldsymbol{\lambda} \circ \left(\mathbf{v} - \mathbf{w}\right)\right)^2 = \sum_{i=1}^{n} \lambda_i^2 \cdot \left(v_i - w_i\right)^2. \tag{2.49}$$

In (2.49) the symbol $\circ$ is the Hadamard product and $\boldsymbol{\lambda}$ is the relevance vector consisting of relevance weights λ_i. Frequently the relevances are normalized such that $\sum_{t=1}^{n} \lambda_i^2 = 1$ is valid to prevent the learning algorithm from degeneration. The associated cost function reads as:

$$E_{GRLVQ} = \frac{1}{2 \cdot N_V} \sum_{\mathbf{v} \in V} f\left(\mu_{\boldsymbol{\lambda}}\left(\mathbf{v}\right)\right), \tag{2.50}$$

with $\mu_{\boldsymbol{\lambda}}(\mathbf{v}) = \frac{d^+_{E,\boldsymbol{\lambda}}(\mathbf{v}) - d^-_{E,\boldsymbol{\lambda}}(\mathbf{v})}{d^+_{E,\boldsymbol{\lambda}}(\mathbf{v}) + d^-_{E,\boldsymbol{\lambda}}(\mathbf{v})}$. For minimizing the cost function the parameters to be optimized are the prototypes as well as the relevance vector, which is realized again by SGD. The prototypes as well as the relevance vector are updated simultaneously, i. e. for a randomly selected training data point $\mathbf{v}$ with label $x(\mathbf{v})$ the prototype adaption is according to

$$\triangle \mathbf{w}^{\pm} \sim \frac{\partial f}{\partial \mu_{\boldsymbol{\lambda}}} \cdot \frac{\partial \mu_{\boldsymbol{\lambda}}}{\partial d^{\pm}_{E,\boldsymbol{\lambda}}(\mathbf{v})} \cdot \frac{\partial d^{\pm}_{E,\boldsymbol{\lambda}}(\mathbf{v})}{\partial \mathbf{w}^{\pm}} \tag{2.51}$$

with

$$\frac{\partial d^{\pm}_{E,\boldsymbol{\lambda}}(\mathbf{v})}{\partial \mathbf{w}^{\pm}} = -2\boldsymbol{\lambda}^2 \circ \left(\mathbf{v} - \mathbf{w}^{\pm}\right)$$

and the relevance vector $\boldsymbol{\lambda}$ is updated by

$$\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} - \eta_{\boldsymbol{\lambda}} \cdot \triangle \boldsymbol{\lambda} \tag{2.52}$$

with

$$\triangle \boldsymbol{\lambda} \sim -\frac{\partial f}{\partial \mu_{\boldsymbol{\lambda}}} \cdot \left(\frac{\partial \mu_{\boldsymbol{\lambda}}}{\partial d^{+}_{E,\boldsymbol{\lambda}}(\mathbf{v})} \cdot \frac{\partial d^{+}_{E,\boldsymbol{\lambda}}(\mathbf{v})}{\partial \boldsymbol{\lambda}} + \frac{\partial \mu_{\boldsymbol{\lambda}}}{\partial d^{-}_{E,\boldsymbol{\lambda}}(\mathbf{v})} \cdot \frac{\partial d^{-}_{E,\boldsymbol{\lambda}}(\mathbf{v})}{\partial \boldsymbol{\lambda}} \right), \tag{2.53}$$

and

$$\frac{\partial d_{E,\boldsymbol{\lambda}}(\mathbf{v})}{\partial \boldsymbol{\lambda}} = 2\boldsymbol{\lambda} \circ (\mathbf{v} - \mathbf{w})^2 .$$

The parameter $\eta_{\boldsymbol{\lambda}}$ in (2.52) is the learning rate for relevance vector. For this leaning rate the same conditions are valid as for $\eta_{\mathbf{w}}$, i. e. $0 < \eta_{\boldsymbol{\lambda}} \ll 1$ and $\eta_{\boldsymbol{\lambda}}$ has to be decreased during learning, see Appendix A. The learning rates $\eta_{\boldsymbol{\lambda}}$ and $\eta_{\mathbf{w}}$ can be initialized independently of each other.
After training the squared relevance weights λ_i^2 reflect the importance of the different dimensions for classification. This means irrelevant dimensions get small weight values but not vice versa. If single features are be dependent of each other and a feature with high λ_i^2 is dropped, other features may be able to compensate the omission [40]. Note that, in GRLVQ the relevances λ_i weight each data dimension independently of each other. If the relevance vector $\boldsymbol{\lambda}^2$ is considered as diagonal matrix, i. e.

$$\boldsymbol{\lambda}^2 = \begin{pmatrix} \lambda_1^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_n^2 \end{pmatrix},$$

the relevance profile can be interpreted as linear mapping, which only causes a scaling of the axes. A further extension of this learning algorithm, subsequently explained, treats the data dimensions no longer independently and, hence, is able to recognize alternative directions in feature space realizing a more discriminative power to classify the data [119].

Matrix Learning in Generalized LVQ

Generalized Matrix Learning Vector Quantization (GMLVQ), proposed by SCHNEIDER ET AL., takes into account pairwise correlations between data dimensions, which is realized by the full matrix $\mathbf{\Lambda}$ in distance measure instead of the relevance vector $\boldsymbol{\lambda}^2$, i. e. the quadratic form

$$d_{E,\mathbf{\Lambda}}(\mathbf{v},\mathbf{w}) = (\mathbf{v}-\mathbf{w})^\top \Lambda (\mathbf{v}-\mathbf{w}), \tag{2.54}$$

with $\mathbf{\Lambda} \in \mathbb{R}^{n\times n}$ as distance is applied [119]. The dissimilarity measure $d_{E,\mathbf{\Lambda}}(\mathbf{v},\mathbf{w})$ defines a generalized squared Euclidean distance in the correspondingly transformed space only if $\mathbf{\Lambda}$ is positive semi-definite and symmetric. In order to satisfy this a parameterization

$$\mathbf{\Lambda} = \mathbf{\Omega}^\top \mathbf{\Omega},$$

is applied, where $\mathbf{\Omega} \in \mathbb{R}^{m\times n}$ is an arbitrary matrix. This yields

$$(\mathbf{v}-\mathbf{w})^\top \Lambda (\mathbf{v}-\mathbf{w}) = (\mathbf{v}-\mathbf{w})^\top \mathbf{\Omega}^\top \mathbf{\Omega} (\mathbf{v}-\mathbf{w}) = \left(\mathbf{\Omega}^\top (\mathbf{v}-\mathbf{w})\right)^2 \geq 0.$$

$\mathbf{\Omega}$ defines a linear mapping of data and prototypes to a new feature space of dimensionality in which, hence, the standard squared Euclidean distance is valid:

$$d_{E,\mathbf{\Omega}}(\mathbf{v},\mathbf{w}) = (\mathbf{\Omega}\cdot(\mathbf{v}-\mathbf{w}))^2 = \sum_{i=1}^{m}\left(\sum_{j=1}^{n}\Omega_{i,j}(v_j-w_j)\right)^2. \tag{2.55}$$

The GMLVQ cost function is defined as:

$$E_{GMLVQ} = \frac{1}{2\cdot N_V}\sum_{\mathbf{v}\in V} f(\mu_{\mathbf{\Omega}}(\mathbf{v})), \tag{2.56}$$

with $\mu_{\mathbf{\Omega}}(\mathbf{v}) = \frac{d^+_{E,\mathbf{\Omega}}(\mathbf{v})-d^-_{E,\mathbf{\Omega}}(\mathbf{v})}{d^+_{E,\mathbf{\Omega}}(\mathbf{v})+d^-_{E,\mathbf{\Omega}}(\mathbf{v})}$. Analogously to GRLVQ, the prototypes $\mathbf{w}^\pm$ and the matrix $\mathbf{\Omega}$ are optimized in parallel by the stochastic derivatives $\frac{\partial_S E_{GMLVQ}}{\partial \mathbf{w}^\pm}$ and

$\frac{\partial_S E_{GMLVQ}}{\partial \Omega_{i,j}}$. Then, the adaptation for the prototypes is according to

$$\triangle \mathbf{w}^{\pm} \sim \frac{\partial f}{\partial \mu_{\mathbf{\Omega}}} \cdot \frac{\partial \mu_{\mathbf{\Omega}}}{\partial d^{\pm}_{E,\mathbf{\Omega}}(\mathbf{v})} \cdot \frac{\partial d^{\pm}_{E,\mathbf{\Omega}}(\mathbf{v})}{\partial \mathbf{w}^{\pm}} \tag{2.57}$$

with

$$\frac{\partial d^{\pm}_{E,\mathbf{\Omega}}(\mathbf{v})}{\partial \mathbf{w}^{\pm}} = -2\left(\mathbf{\Omega} \cdot \left(\mathbf{v} - \mathbf{w}^{\pm}\right)\right).$$

The matrix entries $\Omega_{i,j}$ are adapted by

$$\mathbf{\Omega} \leftarrow \mathbf{\Omega} - \eta_{\mathbf{\Omega}} \cdot \triangle \mathbf{\Omega} \tag{2.58}$$

with

$$\triangle \mathbf{\Omega} = -\frac{\partial f}{\partial \mu_{\mathbf{\Omega}}} \cdot \left(\frac{\partial \mu_{\mathbf{\Omega}}}{\partial d^{+}_{E,\mathbf{\Omega}}(\mathbf{v})} \cdot \frac{\partial d^{+}_{E,\mathbf{\Omega}}(\mathbf{v})}{\partial \Omega_{i,j}} + \frac{\partial \mu_{\mathbf{\Omega}}}{\partial d^{-}_{E,\mathbf{\Omega}}(\mathbf{v})} \cdot \frac{\partial d^{-}_{E,\mathbf{\Omega}}(\mathbf{v})}{\partial \Omega_{i,j}} \right), \tag{2.59}$$

and the derivative

$$\frac{\partial d_{E,\mathbf{\Omega}}(\mathbf{v})}{\triangle \mathbf{\Omega}} = -2\left(\mathbf{\Omega}\left(\mathbf{v} - \mathbf{w}^{\pm}\right)\right), \tag{2.60}$$

where $\eta_{\mathbf{\Omega}}$ is the learning rate for the matrix entries $\Omega_{i,j}$ for which the same conditions are valid as for $\eta_{\boldsymbol{\lambda}}$. After every update step, $\mathbf{\Omega}$ has to be normalized to prevent the learning algorithm from degeneration [120]. The possibilities are to set trace $(\mathbf{\Lambda})$ or det $(\mathbf{\Lambda})$ to a fixed value, hence, either the sum of eigenvalues or the product of eigenvalues is constant. After learning, those data dimensions in $\mathbf{\Lambda}$ are combined, which supports the class separabilities. Thus, $\mathbf{\Lambda}$ can be interpreted as a *classification correlation matrix* [64], i. e. Λ_{ij} describes the correlation between data features supporting the classification. Further, the diagonal elements Λ_{ii} can be seen as relevance weights comparable to the entries λ_i of the relevance vector $\boldsymbol{\lambda}$ in GRLVQ.
Further, the choice of $m < n$ in $\mathbf{\Omega}$ implies that the classifier is restricted to a reduced number of features compared to the original input dimensionality of the data. In the course of these, the linear mapping can be used to obtain low-dimensional representations of high-dimensional data independence of the classification task, which are different from principal component projection [21]. An interesting aspect of matrix relevance learning is that the resulting matrices Λ is dominated by one or very few eigenvectors of data covariance matrix corresponding to the largest eigenvalues. This is due to the matrix update (2.60), which corresponds to a specific Von Mises iteration neglecting the normalization as outlined in [15].

Chapter 3

General Inner Products, Distance and Dissimilarity Measures for Non-Euclidean Spaces

This chapter introduces the mathematical fundamentals to consider Hebbian Learning and Learning Vector Quantization in non-Euclidean spaces. First, this chapter provides the foundations of semi-inner products (SIPs) in Banach spaces and generalized SIPs. As a widely known example the Minkowski norms and their SIPs are introduced. Additional functional norms and their respective SIPs are also considered. In particular, a SIP for Sobolev spaces is proposed. Further a functional measure the $\boldsymbol{L_p^{TS}}$ -Measure by Lee & Verleysen is investigated. Another alternative for the Euclidean inner product are kernels. Hence, inner products and SIPs for general kernel spaces are introduced.
Considering vector spaces of matrices, which are more general vector spaces, the focus has to shift to matrix norms. In particular, the Schatten-p-norm as well as the **QR**-norm, which are both matrix norms will be investigated and an SIP for these vector spaces is developed, which generates the respective norms. In this chapter the theoretical results, which were developed in the context of this work, are highlighted in boxes. The chapter can be skipped if the mathematical details are not of particular

interest to the reader.

The section is based on

*M. Lange and M. Biehl and T. Villmann, "*Non-Euclidean Principal Component Analysis by Hebbian Learning*", Neurocomputing 147 (2015), pp. 107-119[83]*
*M. Biehl, M. Kästner, M. Lange and T. Villmann, "*Non-Euclidean Principal Component Analysis and Oja's Learning Rule - Theoretical Aspects*", in P.A. Estevez and J.C. Principe and P. Zegers, ed., Advances in Self-Organizing Maps: 9th International Workshop WSOM 2012 Santiage de Chile vol. 198, (Berlin: Springer, 2013), pp. 23-34.[14]*

3.1 Semi-Inner Products on Banach Spaces

Semi-inner product spaces as introduced by LUMER in [88] can be seen as a generalization of inner-product spaces. According to LUMER a semi-inner product is defined as follows:

Definition 3.1. *Let V be a vector space. A semi-inner product (SIP) $[\bullet,\bullet]$ of V is a map $[\bullet,\bullet] : V \times V \to \mathbb{C}$ and the following properties $\forall \mathbf{x},\mathbf{y},\mathbf{z} \in V$ hold:*

1. $[\mathbf{x},\mathbf{x}] > 0$ *and* $[\mathbf{x},\mathbf{x}] = 0$ *for* $\mathbf{x} = \mathbf{0}$ *(positive definite)*
2. $\tau\cdot[\mathbf{x},\mathbf{z}]+[\mathbf{y},\mathbf{z}] = [\tau\cdot\mathbf{x}+\mathbf{y},\mathbf{z}]$ *for* $\tau \in \mathbb{C}$ *(linear with respect to the first argument)*
3. $|[\mathbf{x},\mathbf{y}]|^2 \leq [\mathbf{x},\mathbf{x}]\,[\mathbf{y},\mathbf{y}]$ *(Cauchy-Schwarz inequality)*

A vector space V with a SIP is called a semi-inner product space.

Immediately, it can be observed from this definition that SIPs are not necessarily symmetric, i. e. $[\mathbf{x},\mathbf{y}] \neq \overline{[\mathbf{y},\mathbf{x}]}$. The imposition of a *homogeneity* property adds convenient structure without causing any essential restriction of the SIP [35]. A SIP space V has the homogeneity property when the SIP satisfies

$$[\mathbf{x},\tau\cdot\mathbf{y}] = \bar{\tau}\cdot[\mathbf{x},\mathbf{y}]\,,\ \forall \mathbf{x},\mathbf{y} \in V, \tau \in \mathbb{C} \tag{3.1}$$

with $\bar{\tau}$ being the conjugate complex of τ [35]. All normed vector spaces can be represented as SIP spaces with the homogeneity property [35, 88]. Thus, definition 3.1 implies homogeneity property [35]. In the following, it will be assumed that all

SIP spaces possess the homogeneity property.
Note that, SIPs always find $\mathbf{x}, \mathbf{y}, \mathbf{z} \in V$ such that

$$[\mathbf{x}, \mathbf{y} + \mathbf{z}] \neq [\mathbf{x},\mathbf{y}] + [\mathbf{x},\mathbf{z}]$$

is valid, which is equivalent to the symmetry condition: $[\mathbf{x}, \mathbf{y}] \neq \overline{[\mathbf{y},\mathbf{x}]}$. Further, a SIP space has considerable structure when it possesses a *continuity property* on the right-hand member of the SIP. An SIP space is called continuous if for $\tau \in \mathbb{R}$ and for every $\mathbf{x}, \mathbf{y} \in V$ the real part $\Re$ of the SIP fulfills

$$\lim_{\tau \to 0} \Re\left([\mathbf{y}, \mathbf{x} + \tau \cdot \mathbf{y}]\right) = \Re\left([\mathbf{y}, \mathbf{x}]\right). \tag{3.2}$$

The space is a uniform continuous SIP space, if the limit (3.2) is approached uniformly on $V \times V$. Obviously, the following remark is valid:

Remark 3.2. *A real SIP with* $[\bullet, \bullet] : V \times V \to \mathbb{R}$ *is immediately continuous due to the Cauchy-Schwarz inequality. In particular, the real SIP is also linear in the second argument. Hence, in that case*

$$\mathscr{F}_{\mathbf{x}}[\mathbf{y}] = [\mathbf{x}, \mathbf{y}] \cdot \mathbf{x} \tag{3.3}$$

defines a linear operator.

According to G. Lumer [88] for every normed space and, hence, for each Banach space $\mathcal{B}$, one can construct at least one SIP $[\bullet, \bullet]_{\mathcal{B}}$ consistent with the norm such that

$$\|\mathbf{x}\|_{\mathcal{B}} = \sqrt{[\mathbf{x}, \mathbf{x}]_{\mathcal{B}}} \tag{3.4}$$

is valid. Thus, the SIPs for Banach spaces are generalizations of inner products for Hilbert spaces [102]. In general SIPs are not unique in contrast to inner products. However, for real SIPs the uniqueness is satisfied, see the following explanations, which require the *Fréchet- and Gâteaux- differentiability*:

The norm $\|\bullet\|_{\mathcal{B}}$ is called Gâteaux-differentiable if the limit

$$D_{\mathcal{B}}(x) = \lim_{\tau \to 0} \frac{\|\mathbf{x} + \tau \mathbf{y}\|_{\mathcal{B}} + \|\mathbf{x}\|_{\mathcal{B}}}{\tau}$$

exists. If the limit converges uniformly, $\|\bullet\|_{\mathcal{B}}$ is denoted as uniformly Fréchet-differentiable. J.R. Giles has shown that in case of existence the relation

$$D_{\mathcal{B}}(x) = \frac{\Re\left([\mathbf{y},\mathbf{x}]_{\mathcal{B}}\right)}{\|\mathbf{x}\|_{\mathcal{B}}}$$

is valid [35]. Hence the following remark can be indicated explicitly [146]:

Remark 3.3. *If the norm* $\|\mathbf{x}\|_{\mathcal{B}} = \sqrt{[\mathbf{x},\mathbf{x}]_{\mathcal{B}}}$ is *Gâteaux-differentiable then the respective SIP is unique.*

The continuity of the SIP can be related to the differentiability of the respective norm [35]:

Lemma 3.4. *The SIP* $[\bullet,\bullet]_{\mathcal{B}}$ *is continuous (uniformly continuous) iff the respective norm* $\|\mathbf{x}\|_{\mathcal{B}} = \sqrt{[\mathbf{x},\mathbf{x}]_{\mathcal{B}}}$ is *Gâteaux-differentiable (uniformly Fréchet-differentiable).*

Therefore it can be concluded for real SIPs, as mentioned above:

Corollary 3.5. *If the SIP* $[\bullet,\bullet]_{\mathcal{B}}$ *is continuous or uniformly continuous then it is also unique.*

According to Corollary 3.5, real SIPs are unique.

Further, like for Hilbert spaces, a representer theorem for continuous linear functionals on uniform convex Banach spaces can be formulated, which requires the definition of a uniform convex Banach space [35]:

Definition 3.6. A Banach space with norm $\|\bullet\|_{\mathcal{B}}$ is uniform convex if for each $\varepsilon > 0$ there is a $\delta > 0$ such that $\|\mathbf{x}+\mathbf{y}\|_{\mathcal{B}} \leq 2-\delta$ holds if $\|\mathbf{x}-\mathbf{y}\|_{\mathcal{B}} \geq \varepsilon$ is valid.

The representer theorem for continuous linear functionals on a uniform convex Banach space reads now as [35]:

Proposition 3.7. *Let* $\mathcal{B}$ *be a uniform convex and uniform Fréchet-differentiable Banach space. Let* f *be a linear function, i.e.* $f \in \mathcal{B}^*$. *Then there exists a unique* $\mathbf{y} \in \mathcal{B}$ *such that* $f(\mathbf{x}) = [\mathbf{x},\mathbf{y}]_{\mathcal{B}}$.

3.1.1 Generalized Semi-Inner Products

The generalizations of a SIP spaces suggested by Nath in [92, 93] are given by slight modifications of the definition (3.1) for SIP spaces. More precisely, the Cauchy-Schwarz inequality is replaced by the more general Hölder inequality:

$$|[\mathbf{x},\mathbf{y}]| \leq [\mathbf{x},\mathbf{x}]^{\frac{1}{p}} [\mathbf{y},\mathbf{y}]^{\frac{1}{q}},$$

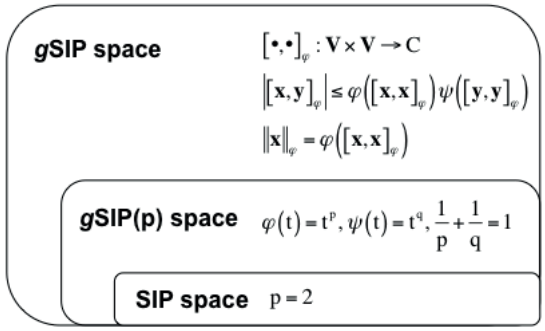

Figure 3.1: Inclusion relations between some kinds of generalized semi-inner product spaces.

where $p, q \in [1, \infty)$ are a pair of conjugated numbers, i. e. $\frac{1}{p} + \frac{1}{q} = 1$ is valid. The respective SIP is denoted as *generalized SIP of type p* (gSIP(p)). For $p = 2$, a gSIP(p) space is reduced to a SIP space. It turns out that also the gSIP(p) space is a normed linear space with $\|\mathbf{x}\| = [\mathbf{x}, \mathbf{x}]^{\frac{1}{p}}$. This result was further extended by Zhang in [146]: More general is the *generalized SIP* (gSIP), which can be defined by a function $[\bullet, \bullet]_\varphi : V \times V \to \mathbb{C}$ on a vector space V satisfying the following three conditions:

1. $[\mathbf{x}, \mathbf{x}]_\varphi > 0$ for all $\mathbf{x} \in V \setminus \{\mathbf{0}\}$ *(Positivity)*
2. $\tau \cdot [\mathbf{x}, \mathbf{z}] + \beta [\mathbf{y}, \mathbf{z}] = [\tau \cdot \mathbf{x} + \beta \mathbf{y}, \mathbf{z}]$ for all $\tau, \beta \in \mathbb{C}$ *and* $\mathbf{x}, \mathbf{y}, \mathbf{z} \in V$ *(linear with respect to the first argument)*
3. For some $\varphi, \psi : \mathbb{R}_+ \to \mathbb{R}_+$ holds $\left|[\mathbf{x}, \mathbf{y}]_\varphi\right| \le \varphi\left([\mathbf{x}, \mathbf{x}]_\varphi\right) \psi\left([\mathbf{y}, \mathbf{y}]_\varphi\right)$ for all $\mathbf{x}, \mathbf{y} \in V$ and the equality holds when $\mathbf{x} = \mathbf{y}$. *(generalized Cauchy-Schwarz inequality)*

The gSIP reduces to gSIP(p) if $\varphi(t) = t^p$ and $\psi(t) = t^q$ is taken, where $p, q \in (1, \infty)$ with $\frac{1}{p} + \frac{1}{q} = 1$. Obviously, if $[\bullet, \bullet]_\varphi$ is a gSIP on a vector space V then $\|\mathbf{x}\|_\varphi = \varphi\left([\mathbf{x}, \mathbf{x}]_\varphi\right)$ defines a norm on V. Conversely, for any normed vector space V exists a gSIP if the map is surjective on $\mathbb{R}_+$ [146]. The relations between the generalized SIP spaces are illustrated in Figure (3.1).

The section is based on

*M. Lange, M. Biehl and T. Villmann, "*Non-Euclidean Principal Component Analysis by Hebbian Learning*", Neurocomputing 147 (2015), pp. 107-119.[83]*

3.2 Minkowski Norms and their Semi-Inner Products

The prominent metric in l_p-spaces is the Minkowski-metric with the l_p-norm

$$\|\mathbf{x}\|_{l_p} = \sqrt[p]{\sum_{i=1}^{n} |x_i|^p} \tag{3.5}$$

for $p \in [1, \infty]$ and $\mathbf{x}, \mathbf{y} \in \mathbb{C}^n$. The respective distance

$$d_{l_p}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_{l_p} \tag{3.6}$$

is known as Minkowski distance. Depending on the selected p-value the Minkowski distance (3.6) displays different behaviors. Great variations in a single dimension become important for large p-values. For $p < 1$ small variations are emphasized and the corresponding unit circle becomes concave, see Figure 3.2.
The norm of the l_p-space (Banach space) is Gâteaux-differentiable.[1] The respective unique and continuous SIPs is

$$[\mathbf{x}, \mathbf{y}]_{l_p} = \frac{1}{\left(\|\mathbf{y}\|_{l_p}\right)^{p-2}} \sum_{i=1}^{n} x_i \bar{y}_i \, |y_i|^{p-2} \,. \tag{3.7}$$

which reduces to

$$[\mathbf{x}, \mathbf{y}]_{l_p} = \frac{1}{\left(\|\mathbf{y}\|_{l_p}\right)^{p-2}} \sum_{i=1}^{n} x_i \, |y_i|^{p-1} \, \mathsf{sign}\,(y_i) \tag{3.8}$$

for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. In (3.8) the term $\mathsf{sign}\,(\cdot)$ denotes the already introduced signum function. The most familiar examples are l_1-norm, l_2-norm and l_∞-norm equipped wither (semi)-inner products and respective distance are summarized in Table 3.1.

[1] Note that, the norm of the $\hat{l_p}$-spaces looks identical to l_p-spaces because of the definition of the absolute value of complex numbers.

p-values	norm	(semi-) inner product	distance
$\mathbf{p=1}$	$\|\mathbf{x}\|_{l_1}=\sum_{i=1}^{n}\|x_i\|$	$[\mathbf{x},\mathbf{y}]_{l_1}=\|\mathbf{y}\|_{l_1}\sum_{i=1}^{n}x_i\cdot\frac{y_i}{\|y_i\|}$	*Manhatten distance* $d_{l_1}(\mathbf{x},\mathbf{y})=\sum_{i=1}^{n}\|x_i-y_i\|$
$\mathbf{p=2}$	$\|\mathbf{x}\|_{l_2}=\sqrt{\sum_{i=1}^{n}x_i^2}$	$[\mathbf{x},\mathbf{y}]_{l_2}=\sum_{i=1}^{n}x_iy_i$	*Euclidean distance* $d_{l_2}(\mathbf{x},\mathbf{y})=\sqrt[2]{\sum_{i=1}^{n}\|x_i-y_i\|^2}$
$\mathbf{p=\infty}$	$\|\mathbf{x}\|_{l_\infty}=\sup_i\|x_i\|$	not known	*maximum distance* $d_\infty(\mathbf{x},\mathbf{y})=\max_i\|x_i-y_i\|$

Table 3.1: The l_p-norms for $p\in\{1,2,\infty\}$ with their (semi-) inner products and the respective distance are listed.

Closely related to the l_p-spaces are the Banach spaces $\mathcal{L}_p$ of complex Lebesgue-integratable functions. For the complex functions g and f the $\mathcal{L}_p$ -spaces are equipped with the SIP

$$[f,g]_{\mathcal{L}_p}=\frac{1}{\left(\|g\|_{\mathcal{L}_p}\right)^{p-2}}\int f\cdot\bar{g}\,|g|^{p-2}\,dt. \tag{3.9}$$

Their real counterparts are the spaces $\mathcal{L}_p$ of real Lebesgue-integratable functions. The SIP (3.9) reduces to

$$[f,g]_{\mathcal{L}_p}=\frac{1}{\left(\|g\|_{\mathcal{L}_p}\right)^{p-2}}\int f\cdot|g|^{p-1}\,\mathsf{sign}\,(g)\,dt, \tag{3.10}$$

where g and f are real functions, generates the functional $\mathcal{L}_p$-norm

$$\|x\,(t)\|_{\mathcal{L}_p}=\int|x\,(t)|^p\,dt.$$

It is well known, that l_p-spaces and $\mathcal{L}_p$-spaces are uniform convex for $p\in(1,\infty)$, see definition (3.6).

General Notes on l_p-norms

The expression in (3.5) is still well-defined for $0<p<1$. However, it is no longer a norm and reduces to be a *quasi-norm* [100]. For a general quasi-norm $\widehat{\|\cdot\|}$ a relaxed triangle inequality

$$\widehat{\|\mathbf{v}\|}+\widehat{\|\mathbf{w}\|}\leq C\widehat{\|\mathbf{v}+\mathbf{w}\|}$$

holds with a quasi-norm constant $C\geq 1$. The other norm axioms are still fulfilled. For the quasi-norms $\|\cdot\|_{l_p}$ with $p\in(0,1)$, this constant is obtained as $C=\max\left\{1,2^{\frac{p}{1-p}}\right\}$

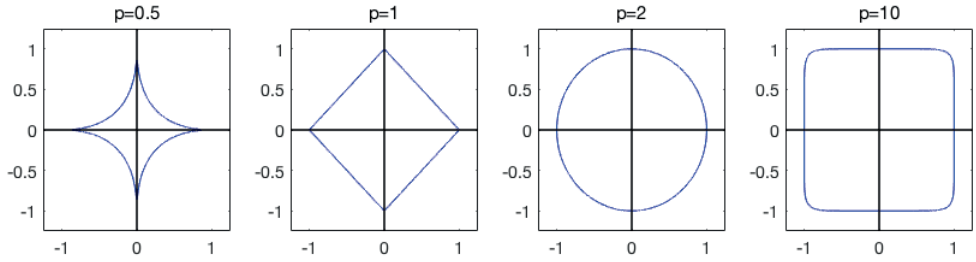

Figure 3.2: Unit circles for several l_p-norms (from left to right): $p = \frac{1}{2}$, $p = 1$ (called Manhatten distance), $p = 2$ (called Euclidean distance) and $p = 10$.

and the respective vector space is a complete Quasi-Banach space [31]. Additionally, the *Minkowski inequality*

$$\| |\mathbf{v}| + |\mathbf{w}| \|_{l_p} \leq \|\mathbf{v}\|_{l_p} + \|\mathbf{w}\|_{l_p}$$

holds for those l_p-quasi-norms with $|\mathbf{x}| = (|x_1|, \ldots, |x_n|)^\top$. Further, the *p-triangle inequality*

$$\|\mathbf{v}\|_{l_p}^p + \|\mathbf{w}\|_{l_p}^p \leq \|\mathbf{v} + \mathbf{w}\|_{l_p}^p$$

is valid. It turns out that

$$d_{l_p}(\mathbf{v}, \mathbf{w}) = \left(\|\mathbf{v} - \mathbf{w}\|_{l_p} \right)^p$$

is a translation invariant metric also for the quasi-norm case [58]. Note that here, the l_p-space with $d_{l_p}(\mathbf{v}, \mathbf{w})$ is a so called *Fréchet-space* (F-space) and not a Banach space. F-spaces are generalizations of Banach spaces. Thus each Banach space is also an F-space. F-spaces are complete with respect to the metric, the metric is translation invariant and continuous. Yet, F-spaces are not locally convex whereas Banach spaces fulfill this property [41].

The sections are based on

M. Lange, M. Biehl and T. Villmann, "Non-Euclidean Principal Component Analysis by Hebbian Learning", *Neurocomputing 147 (2015), pp. 107-119. [83]*
T. Villmann and M. Lange, "A comment on the functional L_p^{TS}-Measure Regarding the norm properties", *TechReport, 2015.[140]*

3.3 Functional Norms and their Semi-Inner Products Based on Minkowski-Norms

Although $\hat{\mathcal{L}}_p$ and $\mathcal{L}_p$ are function spaces, the respective norms and SIP's do not involve the functional character explicitly. More precise, the values are invariant under transformation of the function such that the function values for two arbitrary arguments t_1 and t_2 is switched. The same property is observed for discrete versions if vector dimensions are switched. For this reasons so called functional norms will be introduced [105]. Especially the Sobolev spaces are of great interest in functional data analysis. Of particular interest in the context of this work is the SIP of the Sobolev space, which can be applied for Hebbian approaches in non-Euclidean spaces. Subsequently, an alternative discrete functional norm, the Functional $\boldsymbol{L_p^{TS}}$ -Measure introduced by Lee and Verleysen, will be considered. Here, it will be shown, that the $\boldsymbol{L_p^{TS}}$ -Measure defines only a quasi-norm, because the triangle inequality is violated [140].

3.3.1 The Sobolev Spaces

Closely related to the real $\mathcal{L}_p$-space is the Sobolev-space
$\mathcal{W}_{K,p} = \{f \mid D^\alpha f \in \mathcal{L}_p, |\alpha| \leq K,\, K \in \mathbb{N}\}$ of real differentiable functions up to order K with $D^\alpha = \frac{\partial^{|\alpha|}}{\partial\alpha_1 \ldots \partial\alpha_{|\alpha|}}$ being the differential operator of order $|\alpha|$. The norm of $\mathcal{W}_{K,p}$ is defined by

$$\begin{aligned} \|f\|_{\mathcal{W}_{K,p}} &:= \left(\sum_{|\alpha|\leq K} \|D^\alpha f\|_{\mathcal{L}_p}^p \right)^{\frac{1}{p}} \\ &= \left(\sum_{|\alpha|\leq K} \int |D^\alpha f|^p \, dt \right)^{\frac{1}{p}}, \end{aligned} \tag{3.11}$$

which is based on the $\mathcal{L}_p$-norm. It is well known, that $\mathcal{W}_{K,p}$ and $\mathcal{L}_p$ are Hilbert spaces only for $p = 2$ [1]. Similar to the SIP (3.10) of the $\mathcal{L}_p$-space the unique SIP of $\mathcal{W}_{K,p}$ can be defined as:

Lemma 3.8. (Definition of SIP in $\mathcal{W}_{K,p}$) *The unique SIP of $\mathcal{W}_{K,p}$ is given as*

$$[f,g]_{\mathcal{W}_{K,p}} := \frac{1}{\|g\|^{p-2}_{\mathcal{W}_{K,p}}} \sum_{|\alpha|\leq K} \int D^\alpha f \, |D^\alpha g|^{p-1} \, \mathsf{sgn}\left(D^\alpha g\right) dt. \tag{3.12}$$

The proof of the SIP properties and uniqueness can be found in Appendix (B.1) based on [83]. The differentiability of the functions g and f is obviously required in (3.11) and (3.12). If the differentiability property cannot be ensured, this might be a disadvantage. However, in most cases in data mining only discrete approximations of functions are considered. That means, $\mathbf{x} \in \mathbb{R}^n$ are discrete representations of functions and their vector entries x_k and x_{k+j} are functionally related depending on the index difference j. Consequently, this functional aspect of the vector entries is lost if only the l_p-norm (3.5) is used. Otherwise, machine learning algorithms in data mining may benefit from those functional data properties [112, 113, 143].

3.3.2 The Functional $\boldsymbol{L_p^{TS}}$ -Measure

An alternative discrete functional dissimilarity was introduced by Lee & Verleysen in [84] as a generalization of l_p-norms incorporating functional data properties. The functional structure of data vectors is taken into account by involving the previous and next values of x_i of a vector $\mathbf{x} = (x_0, \ldots, x_{D+1}) \in \mathbb{R}^n$ with $n = D + 2$. In the calculation of the dissimilarity value, as explained in the following, the notation $x_0 = x\,(0)\,,\ x_1 = x\,(1)\,,\ x_2 = x\,(2)\,, \ldots$ is used. Assuming that the sampling period τ is constant and $x_0 = x_{D+1} = 0$ is valid. The proposed dissimilarity measure is defined as

$$\delta_p\left(\mathbf{x},\mathbf{y}\right) = \boldsymbol{L_p^{TS}}\left(\mathbf{x}-\mathbf{y},\tau\right) \tag{3.13}$$

with

$$L_p^{TS}\left(\mathbf{x},\tau\right) = \left(\sum_{i=1}^{D} \left(A_i\left(\mathbf{x},\tau\right) + B_i\left(\mathbf{x},\tau\right)\right)^p\right)^{\frac{1}{p}}, \tag{3.14}$$

where

$$A_i\left(\mathbf{x},\tau\right) = \begin{cases} \frac{\tau}{2}\,|x_i| & 0 \leq x_i x_{i-1} \\ \frac{\tau}{2}\,\frac{x_i^2}{|x_i|+|x_{i-1}|} & 0 > x_i x_{i-1} \end{cases} \tag{3.15}$$

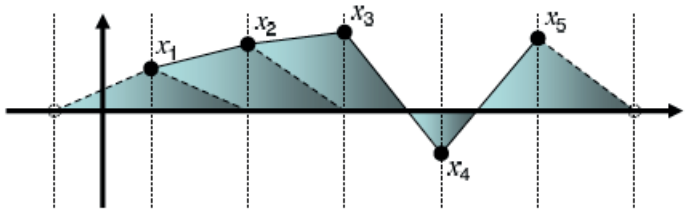

Figure 3.3: Illustration of the quasi-norm $\boldsymbol{L_p^{TS}}$ taken from [84]. The quasi-norm involves the areas of the triangles located on the left A_i and right B_i side of each coordinate.

and

$$B_i(\mathbf{x},\tau) = \begin{cases} \frac{\tau}{2}|x_i| & 0 \leq x_i x_{i+1} \\ \frac{\tau}{2}\frac{x_i^2}{|x_i|+|x_{i+1}|} & 0 > x_i x_{i+1} \end{cases} . \tag{3.16}$$

are the respective areas of triangles on the left and right sides of x_i, as depicted in Figure 3.3. Lee and Verleysen proposed in [84] that the quantity $L_p^{TS}(\mathbf{x},\tau)$ defined in (3.14) is a norm and, therefore, $\delta_p(\mathbf{x},\mathbf{y})$ is assumed to be a distance.

Note 3.9. However, it can be verified that the triangle inequality may be violated, see Appendix B.2. Therefore, $L_p^{TS}(\mathbf{x},\tau)$ is only a quasi-norm and thus also the dissimilarity measure $\delta_p(\mathbf{x},\mathbf{y})$ from (3.13) is only a quasi-metric.

In the following, the approximation properties of the quasi-norm $L_p^{TS}(\mathbf{x},\tau)$ with respect to the functional $\mathcal{L}_p$-norm

$$\|x(t)\|_{\mathcal{L}_p} = \int |x(t)|^p \, dt$$

are briefly investigated (as in publication [140]). For that, the vector $\mathbf{x}$ is supposed to be a discrete representation of a continuous function $x(t)$, as suggested in [84]. Then the difference between two consecutive points corresponds to a small interval Δt scaled by a sampling period τ. The functional counterparts of $A_i(\mathbf{x},\tau)$ and $B_i(\mathbf{x},\tau)$ are defined as

$$\alpha(x(t),t,\tau,\Delta t) = \frac{\tau}{2}|x(t)| \cdot \left(H(x(t)\cdot x(t-\Delta t)) + \frac{1-H(x(t)\cdot x(t-\Delta t))}{1+\frac{|x(t-\Delta t)|}{|x(t)|}} \right), \tag{3.17}$$

and

$$\beta\left(x\left(t\right),t,\tau,\Delta t\right)=\frac{\tau}{2}\left|x\left(t\right)\right|\cdot\left(H\left(x\left(t\right)\cdot x\left(t+\Delta t\right)\right)+\frac{1-H\left(x\left(t\right)\cdot x\left(t+\Delta t\right)\right)}{1+\frac{|x(t+\Delta t)|}{|x(t)|}}\right), \tag{3.18}$$

respectively, where $H\left(x\right)$ is the already known Heaviside function. $A_i\left(\mathbf{x},\tau\right)$ and $B_i\left(\mathbf{x},\tau\right)$ are summed up in $L_p^{TS}\left(\mathbf{x},\tau\right)$. The respective summation in the functional case yields

$$\alpha\left(x\left(t\right),t,\tau,\Delta t\right)+\beta\left(x\left(t\right),t,\tau,\Delta t\right) \tag{3.19}$$

as a $\tau\cdot\Delta t$-dependent counterpart. The term

$$\begin{aligned}\vartheta_x\left(t,\tau,\Delta t\right) &= H\left(x\left(t\right)\cdot x\left(t-\Delta t\right)\right)+H\left(x\left(t\right)\cdot x\left(t+\Delta t\right)\right)\\ &\quad+\frac{1-H(x(t)\cdot x(t-\Delta t))}{1+\frac{|x(t-\Delta t)|}{|x(t)|}}+\frac{1-H(x(t)\cdot x(t+\Delta t))}{1+\frac{|x(t+\Delta t)|}{|x(t)|}}\end{aligned}$$

can be interpreted as a multiplicative deviation of $x\left(t\right)$. Thus, the summation in (3.19) can be rewritten as

$$\frac{\tau}{2}\left|x\left(t\right)\cdot\vartheta_x\left(t,\tau,\Delta t\right)\right| \tag{3.20}$$

by applying the observation $\vartheta_x\left(t,\tau,\Delta t\right)\geq 0$. Although it is assumed that $x\left(t\right)$ is a continuous function, the deviation function $\vartheta_x\left(t,\tau,\Delta t\right)$ is not requisitely continuous everywhere regarding to the difference Δt. This can be illustrated by considering a continuous time-dependent function $x\left(t\right)$ on the interval $[a,b]$ with $x\left(a\right)\cdot x\left(b\right)<0$. Without loss of generality assuming that $x\left(a\right)<0$. Then exists at least one t_0 with $x\left(t_0\right)=0$ together with an $\varepsilon>0$ determining the interval $I_\varepsilon\left(t_0\right)=[t_0-\varepsilon,t_0+\varepsilon]$ such that it fulfills the following statements:

1. $I_\varepsilon\left(t_0\right)\subseteq[a,b]$

2. $x\left(t_0-\varepsilon\right)\cdot x\left(t_0+\varepsilon\right)<0$

3. $x\left(t\right)$ is monotonically increasing in $I_\varepsilon\left(t_0\right)$

4. $x\left(t\right)<0$ for $t\in[t_0-\varepsilon,t_0)$ and $x\left(t\right)>0$ for $t\in(t_0,t_0+\varepsilon]$.

Let $t^*\in(t_0-\varepsilon,t_0)$ be arbitrarily but fixed and $\Delta t=t-t^*<\frac{\varepsilon}{2}$. Thus, the strong inequality $x\left(t_0\right)>x\left(t^*\right)$ holds. The limit $t^*\to t$ of the function $\alpha\left(x\left(t\right),t,\tau,\Delta t\right)$

from (3.17) yields

$$\begin{aligned}\lim_{\Delta t\to 0}\alpha\left(x\left(t_0\right),t_0,\tau,\Delta t\right) &= \lim_{\Delta t\to 0}\frac{\tau}{2}\cdot\frac{\left|x\left(t_0\right)\right|}{1+\frac{\left|x(t_0-\Delta t)\right|}{\left|x(t_0)\right|}} \\ &= \frac{\tau}{4}\cdot\left|x\left(t_0\right)\right| \\ &\neq \alpha\left(x\left(t_0\right),t_0,\tau,0\right)\end{aligned} \tag{3.21}$$

because of $\alpha\left(x\left(t_0\right),t_0,\tau,0\right)=\frac{\tau}{2}\cdot\left|x\left(t_0\right)\right|$ by taking the Heaviside function $H\left(x\right)$ into consideration. The limit value observation of $\beta\left(x\left(t\right),t,\tau,\Delta t\right)$ can be performed in the same way.

Note 3.10. Therefore, the deviation $\vartheta_x\left(t,\tau,\Delta t\right)$ in (3.20) is not requisitely continuous and a continuous approximation of the functional $\mathcal{L}_p$-norm cannot be obtained.

The section is based on

*M. Lange, M. Biehl and T. Villmann, "*Non-Euclidean Principal Component Analysis by Hebbian Learning*", Neurocomputing 147 (2015), pp. 107-119.[83]*

3.4 Further General Notes on Banach Spaces

In the following some additional notes on Banach spaces l_p and $\mathcal{W}_{K,p}$ are presented, which are fundamental for algorithms introduced in the next chapters.

Remark 3.11. In a Banach space $\mathcal{B}$ two vectors $\mathbf{v}$ and $\mathbf{w}$ are considered. The vector $\mathbf{v}$ is normal to the vector $\mathbf{w}$ and the vector $\mathbf{w}$ is transversal to the vector $\mathbf{v}$ iff $\left[\mathbf{v},\mathbf{w}\right]_{\mathcal{B}}=0$, i. e. the orthogonality relation is not symmetric.

In the following some properties regarding the separability of Banach spaces are collected: Note that, the separability property is not sufficient for a countable basis of an infinite dimensional Banach space $\mathcal{B}$. However, the following statement is valid:

Remark 3.12. If a countable set of elements $B_s=\{b_k\in\mathcal{B}\mid k\in\mathbb{N}\}$ exists and B_s is dense in $\mathcal{B}$ then it is called a *Schauder basis*, implying the separability of $\mathcal{B}$ and a respective unique vector representation $\mathbf{v}=\sum_{k=1}^{\infty}v_kb_k$ for all infinite dimensional vectors $\mathbf{v}\in\mathcal{B}$ [58].

The basis is called *unconditional*, if the representation $\mathbf{v}=\sum_{k=1}^{\infty}v_kb_k$ converges unconditionally. Further, the Banach spaces l_p for $p\in[1,\infty)$ with SIP (3.7) and $\mathcal{L}_p\left(\mathcal{K}\right)$ over a compact set $\mathcal{K}\in\mathbb{R}^n$ with SIP (3.7) have a Schauder basis. The same applies to the real counterparts with the SIPs (3.8) and (3.10), respectively.

The latter remark also implies a Schauder basis for the Sobolev space $\mathcal{W}_{K,p}(\mathcal{K})$ with SIP (3.12).

Let $\mathcal{B}^*$ be the dual space of linear functionals over $\mathcal{B}$ with Schauder basis $B_s = \{b_k \in \mathbb{B} \mid k \in \mathbb{N}\}$ and an arbitrary subspace $\mathcal{B}_n \subset \mathcal{B}$ spanned by $b_1, \ldots, b_n$ with $\mathcal{B}_n^*$. Consider a function $f \in \mathcal{B}^*$ and its restriction $f\mid_{\mathbb{B}_n} \in \mathcal{B}^*$. The basis B_s is called *shrinking* if $\lim_{n\to\infty} \|f\mid_{\mathcal{B}_n}\| = 0$ is valid.

Remark 3.13. According to a theorem provided by R.C. James, a Banach space is reflexive iff it has an unconditional shrinking Schauder basis [62]. Hence, a Schauder basis for reflexive Banach spaces can always be supposed.

Note that, these mathematical considerations for SIPs remain also valid for generalized SIPs. This concerns in particular the uniqueness, existence, and approximation capability based on the Schauder basis theory for Banach spaces. More detailed mathematical analysis can be found in [146] and [51].

The section is based on

M. Lange, M. Biehl and T. Villmann, "Non-Euclidean Principal Component Analysis by Hebbian Learning", *Neurocomputing 147 (2015), pp. 107-119.[83]*

3.5 Inner Products and Semi-Inner Products for General Kernel Spaces

Kernel methods have become popular in machine learning [100]. Roughly speaking, kernels are (conditionally) positive definite (cpd) functions of two variables, which can be thought to encode proximity between pairs of vectors. They are originally defined in vector spaces, e. g. based on a feature representation of vectors. Note that, kernels can be interpreted as generalized inner products in a reproducing kernel Hilbert space (RKHS) [100].

In the following, let $(V,\, d_V)$ be a compact metric space with the vector space V being equipped with an arbitrary metric d_V. A kernel is a function κ on V

$$\kappa_\phi : V \times V \to \mathbb{C}, \tag{3.22}$$

if there exists an associated Hilbert space $\mathcal{H}$ and a feature map

$$\Phi : V \ni \mathbf{v} \to \Phi(\mathbf{v}) \in \mathcal{H} \tag{3.23}$$

with

$$\kappa_\phi(\mathbf{v}, \mathbf{w}) = \langle \Phi(\mathbf{v}), \Phi(\mathbf{w}) \rangle_{\mathcal{H}} \tag{3.24}$$

for all $\mathbf{v}, \mathbf{w} \in V$. $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is the inner product in $\mathcal{H}$. $\mathcal{H}$ is also denoted as the feature space of κ_ϕ. Without further restrictions on the kernel κ_ϕ, neither $\mathcal{H}$ nor Φ are unique [132]. Positive definite kernels are of great importance because they *uniquely* correspond to reproducing kernel Hilbert spaces (RKHS) $\mathcal{H}$ in a canonical manner according to the Mercer-theorem [6, 91, 131]. The kernel κ_ϕ is called positive definite if for all finite subsets $V_m \subseteq V$ with cardinality $|V_m| = m$ the Gram-Matrix

$$\mathbf{G}_m = [\kappa_\phi(\mathbf{v}_i, \mathbf{v}_j) : i, j = 1 \dots m] \tag{3.25}$$

is positive semi-definite [6]. The norm $\|\Phi(\mathbf{v})\|_{\mathcal{H}} = \sqrt{\kappa_\phi(\Phi(\mathbf{v}), \Phi(\mathbf{v}))}$ of this RKHS induces a *kernel semi-metric*

$$d_{\kappa_\phi}(\mathbf{v}, \mathbf{w}) = \sqrt{\kappa_\phi(\mathbf{v}, \mathbf{v}) - 2\kappa_\phi(\mathbf{v}, \mathbf{w}) + \kappa_\phi(\mathbf{w}, \mathbf{w})} \tag{3.26}$$

defined by the kernel κ_ϕ [121]. If the feature map Φ is injective the kernel semi-metric becomes a general metric such that

$$d_{\kappa_\phi}(\mathbf{v}, \mathbf{w}) = \|\Phi(\mathbf{v}) - \Phi(\mathbf{w})\|_{\mathcal{H}} \tag{3.27}$$

is valid [130]. Regarding this, d_{κ_ϕ} is also denoted as *kernel induced* metric. A function $f_h : V \to \mathbb{C}$ is induced by κ_ϕ if there exists an element h in the associated Hilbert space $\mathcal{H}$ with $f_h(\mathbf{v}) = \langle h, \Phi(\mathbf{v}) \rangle_{\mathcal{H}}$. A continuous kernel κ_ϕ on a compact metric space (V, d_V) is called *universal*, if the space $\mathcal{F}$ of all functions induced by κ_ϕ is dense in the space $\mathcal{C}(V)$ of continuous functions over V equipped with the maximum norm 3.1 [130]. Steinwart proved that continuous, universal kernels induce the continuity and separability of the corresponding feature map Φ and the image $\mathcal{I}_{\kappa_\phi} = \Phi(V)$ is a subspace of $\mathcal{H}$ [130, 131]. Further Steinwart shows in [130] that (continuous) universal kernels also imply the continuity and injectivity of the map

$$\Psi : (V, d_V) \to \left(V, d_{\kappa_\phi}\right) \tag{3.28}$$

with $d_{\kappa_\phi}(\mathbf{v}, \mathbf{w}) = d_{\mathcal{H}}(\Phi(\mathbf{v}), \Phi(\mathbf{w}))$, where $\left(V, d_{\kappa_\phi}\right)$ is the compact vector space V with kernel induced metric d_{κ_ϕ}. In [138] was shown that $\left(V, d_{\kappa_\phi}\right)$ is isometric and isomorphic to $\mathcal{I}_{\kappa_\phi}$. This correspondence is visualized in Figure (3.4). For a feature mapping space with weaker assumptions an analogous theory can be derived. Zhang et al. consider in [147] reflexive Banach spaces as mapping spaces. In this work, the

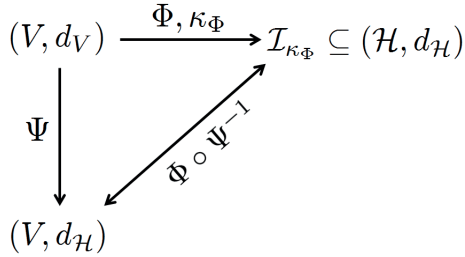

Figure 3.4: Visualization of the statement: For universal kernels κ_ϕ the metric spaces $(V,\, d_V)$ and $\left(\mathcal{I}_{\kappa_\phi},\, d_V\right)$, are topologically equivalent and isometric by means of the continuous bijective mapping $\Phi \circ \Psi^{-1}$ [138].

Banach space is also assumed to be a function space as above for the Hilbert space $\mathcal{H}$. Now, such a reflexive function Banach space $\mathcal{B}$ over the compact metric space $(V,\, d_V)$ with the SIP $[h, g]_\mathcal{B}$ is considered with a reproducing property for Banach spaces, called *Reproducing Kernel Banach space* (RKBS). If the RKBS is Fréchet-differentiable, it is named a *semi-inner product Reproducing Kernel Banach space* (SIP-RKBS). The feature map $\Phi : V \to \mathcal{B}$ is considered again with a so-called SIP-kernel

$$\gamma_\phi\left(\mathbf{v}, \mathbf{w}\right) = \left[\Phi\left(\mathbf{v}\right), \Phi\left(\mathbf{w}\right)\right]_\mathcal{B} \tag{3.29}$$

in $\mathcal{B}$. For a SIP-RKBS $\mathcal{B}$ a unique correspondence exists between a SIP-kernel γ_ϕ and the map Φ based on the Banach space representation theorem [147]. If the map Φ is continuous then also γ_ϕ is. In addition, it can be proved that (weakly) universal SIP-kernels correspond to bijective mappings Φ [138]. Moreover, it follows that the identity map

$$\Psi : (V,\, d_V) \to (V,\, d_\mathcal{B}) \tag{3.30}$$

is also continuous and, therefore, bijective iff the SIP-kernel is (weakly) universal and continuous. Therefore, $(V,\, d_\mathcal{B})$ and the subspace $\mathcal{I}_{\gamma_\phi} = \Phi\left(V\right) \subseteq \mathcal{B}$ are isomorphic and also isometric. These relations are visualized in Figure 3.5 and proven in [138].

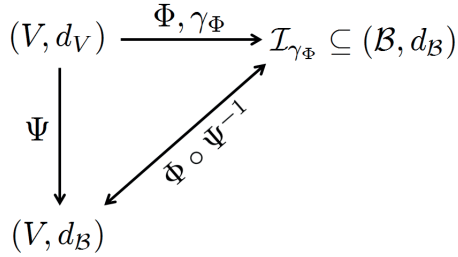

Figure 3.5: Visualization of the statement: For SIP universal kernels γ_ϕ the metric spaces $(V,\ d_\mathcal{B})$ and $\left(\mathcal{I}_{\gamma_\phi},\ d_\mathcal{B}\right)$, are topologically equivalent and isometric by means of the continuous bijective mapping $\Phi \circ \Psi^{-1}$ [138].

The section is based on

K. Domaschke, M. Kaden, M. Lange, T. Villmann, "Learning Matrix Quantization and Variants of Relevance Learning", *in M. Verleysen, ed., Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2015), pp. 13-18, Louvain-La-Neuve, Belgium (2015). [32]*
A. Bohnsack, K. Domaschke, M. Kaden, M. Lange and T. Villmann, "Learning Matrix Quantization and Relevance Learning Based on Schatten-p-norms", *Neurocomputing 192 (2016), pp. 104-114.[19]*
A. Bohnsack, K. Domaschke, M. Kaden, M. Lange and T. Villmann, "Mathematical Characterization of Sophisticated Variants for Relevance Learning in Learning Matrix Quantization Based on Schatten-p-norms", *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science) 1 (2015), pp. 403-414.[18]*
A. Villmann, M. Lange-Geisler, T. Villmann, "About Semi-Inner Products for $\mathbf{p-QR}$-Matrix Norms", *TechReport (2018).[136]*

3.6 Matrix Norms and their Semi-Inner Products

As mentioned in the beginning of this chapter, in this section the Schatten-p-norm and the **QR**-norm as matrix norms will be investigated and the respective generated SIPs will be presented.

3.6.1 Preliminaries

Let $\mathscr{L}(V_m, V_n)$ be the vector space of linear functions between the vector spaces V_m and V_n with dimensionalities n and m, respectively. Hence $\mathscr{L}(V_m, V_n)$ consists of all matrices $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{m \times n}$. If the vector space $\mathscr{L}(V_m, V_n)$ is equipped with an

appropriate norm $\|\bullet\|_{\mathscr{L}}$, it becomes a Banach space $\mathcal{B}_{m,n}$. A norm on matrices that satisfy the usual norm axioms is a *vector norm on matrices*, also called *generalized matrix norm*. However, $\|\bullet\|_{\mathscr{L}}$ is termed simply as *matrix norm,* if the property of *submultiplicativity* is fulfilled additionally, i. e. the Cauchy-Schwarz-inequality $\|\mathbf{A}\cdot\mathbf{B}\|_{\mathscr{L}} \leq \|\mathbf{A}\|_{\mathscr{L}}\cdot\|\mathbf{B}\|_{\mathscr{L}}$ is valid. Sometimes, a matrix norm is called a *ring norm*. The notions *matrix seminorm* and *generalized matrix seminorm* may be defined via omission of *positivity,* i. e. $\|\mathbf{A}\| = 0$ if and only if $\mathbf{A} = \mathbf{0}$. If a vector norm $\|\bullet\|_V$ exists such that

$$\|\mathbf{A}\|_M = \max_{\|\mathbf{v}\|_V=1} \|\mathbf{A}\mathbf{v}\|_V$$

is valid then $\|\bullet\|_M$ is a matrix norm and said to be the *natural norm induced by* the vector norm $\|\bullet\|_V$.

3.6.2 Schatten-p-norms

The definition of Schatten-p-norms $\|\mathbf{A}\|_{\mathcal{S}_p}$ for matrices $\mathbf{A} \in \mathbb{C}^{m\times n}$ uses the trace operator $\mathrm{tr}\,(\cdot)$ and reads as

$$\|\mathbf{A}\|_{\mathcal{S}_p} = \sqrt[p]{\mathrm{tr}\,(|\mathbf{A}|^p)}, \tag{3.31}$$

where $|\mathbf{A}|$ denotes the absolute value of $\mathbf{A}$ with $|\mathbf{A}| = \sqrt{\mathbf{A}^*\mathbf{A}} = \sqrt{\mathbf{A}\mathbf{A}^*}$.[2] Note that the trace operator is linear and cyclic. Hence, $\|\lambda\cdot\mathbf{A}+\gamma\cdot\mathbf{B}\|_{\mathcal{S}_p} = |\lambda|\cdot\|\mathbf{A}\|_{\mathcal{S}_p} + |\gamma|\cdot\|\mathbf{B}\|_{\mathcal{S}_p}$ is valid and

$$\|\mathbf{ABC}\|_{\mathcal{S}_p} = \|\mathbf{CAB}\|_{\mathcal{S}_p} \tag{3.32}$$

holds. This implicates that $\|\mathbf{A}\|_{\mathcal{S}_p}$ is invariant with respect to any basis transformation, i. e. $\|\mathbf{A}\|_{\mathcal{S}_p} = \left\|\mathbf{BAB}^{-1}\right\|_{\mathcal{S}_p}$ for regular $\mathbf{B}$. Further, Schatten-p-norms are by definition also self-adjoint such that $\|\mathbf{A}\|_{\mathcal{S}_p} = \|\mathbf{A}^*\|_{\mathcal{S}_p}$ is valid. Equivalently the Schatten-p-norms can be calculated as

$$\|\mathbf{A}\|_{\mathcal{S}_p} = \sqrt[p]{\sum_{k=1}^{n} (\sigma_k(\mathbf{A}))^p}$$

utilizing the singular values of $\mathbf{A}$ denoted as $\sigma_k(\mathbf{A})$. This observation establishes a relationship to l_p-norms, i. e. $\|\mathbf{A}\|_{\mathcal{S}_p} = \|\sigma(\mathbf{A})\|_{l_p}$ [117]. The following lemma can be stated:

[2] $\mathbf{A}^*$ is the conjugate complex of $\mathbf{A}$.

Lemma 3.14. *The Schatten-p-norm is unitarily invariant, i. e* $\|\mathbf{A}\cdot\mathbf{U}\|_{\mathcal{S}_p} = \|\mathbf{A}\|_{\mathcal{S}_p}$ *with* $\mathbf{U}$ *being an unitary matrix.*

The proof of this theorem is given in Appendix B.4. Because of this invariance property and following theorem in [50, p. 469], Schatten-p-norms belong to matrix norms and thus the Cauchy-Schwarz-inequality

$$\|\mathbf{A}\cdot\mathbf{B}\|_{\mathcal{S}_p} \leq \|\mathbf{A}\|_{\mathcal{S}_p}\cdot\|\mathbf{B}\|_{\mathcal{S}_p} \tag{3.33}$$

is fulfilled.

Special Cases of Schatten-p-norms

The most familiar examples of Schatten-p-norms are $p = 1,\ 2,\ \infty$. For $p = 1$ the Schatten-p-norm reduces to the matrix norm

$$\|\mathbf{A}\|_{\mathcal{S}_1} = \sum_{i,j=1}^{n} |a_{ij}|\,, \tag{3.34}$$

with $\mathbf{A}\in\mathbb{C}^{n\times n}$ and is known as *nuclear or trace norm*. The *Frobenius-norm*, also named *Schur norm* or *Hilbert-Schmidt norm*, results for $p = 2$

$$\|\mathbf{A}\|_{\mathcal{S}_2} = |\mathrm{tr}\,(\mathbf{A}\mathbf{A}^*)|^{\frac{1}{2}} = \left(\sum_{i,j=1}^{n} |a_{ij}|^2\right)^{\frac{1}{2}}, \tag{3.35}$$

which is also a matrix norm. Note that that the Frobenius norm is an absolute norm[3]. The Frobenius norm is the natural norm induced by the Euclidean vector norm. For $p = \infty$ the Schatten-p-norm defines the *spectral norm* and reads as

$$\|\mathbf{A}\|_{\mathcal{S}_\infty} = \max_{1\leq i,j\leq n} |a_{ij}|\,. \tag{3.36}$$

It is a non-submultiplicative norm induced by a vector norm on the vector space $\mathbb{C}^{n\times n}$ but not a matrix norm. Further, the dual of $\|\mathbf{A}\|_{\mathcal{S}_p}$ is the norm $\|\mathbf{A}\|_{\mathcal{S}_q}$ with $\frac{1}{p}+\frac{1}{q} = 1$.

[3] If $\mathbf{x} = [x_i]\in\mathbb{C}^n$, let $|\mathbf{x}| = [|x_1|\ldots|x_n|]$ denote the entry wise absolute value of $\mathbf{x}$. It is $|\mathbf{x}|\leq|\mathbf{y}|$ if $|x_i|\leq|y_i|\ \forall i = 1\ldots n$. A norm $\|\cdot\|$ on $\mathbb{C}^n$ is absolute if $\|\mathbf{x}\| = \||\mathbf{x}|\|\ \forall\mathbf{x}\in\mathbb{C}^n$ holds.

Semi-inner Products for Schatten-p-norms

Commonly known, the Frobenius norm results from the Frobenius inner product

$$[\mathbf{A},\mathbf{B}]_{\mathcal{S}_2} = \mathrm{tr}\left(\mathbf{A}^*\mathbf{B}\right) \tag{3.37}$$

and, hence, the space of matrices becomes a Hilbert space [50]. Thus the Frobenius inner product satisfy the Cauchy-Schwarz inequality, i. e. $\left|\langle\mathbf{A},\mathbf{B}\rangle_{\mathcal{S}_2}\right|^2 \leq \left|\langle\mathbf{A},\mathbf{A}\rangle_{\mathcal{S}_2}\right| \cdot \left|\langle\mathbf{B},\mathbf{B}\rangle_{\mathcal{S}_2}\right|$, and also the Hermitian symmetry. By means of these properties and keeping in mind that Schatten-p-norms correspond to Banach spaces the following proposition can be stated:

Proposition 3.15. *The Banach space $\mathcal{B}_{m,n}$ of matrices $\mathbf{A},\mathbf{B} \in \mathbb{C}^{m\times n}$ equipped with the Schatten-p-norm $\|\mathbf{A}\|_{\mathcal{S}_p} = \sqrt[p]{tr\left(|\mathbf{A}|^p\right)}$ corresponds to the SIP $[\mathbf{A},\mathbf{B}]_{\mathcal{S}_p} : \mathcal{B}_{m,n}\times\mathcal{B}_{m,n}\to\mathbb{C}$ defined as*

$$[\mathbf{A},\mathbf{B}]_{\mathcal{S}_p} = \frac{1}{\left(\|\mathbf{B}\|_{\mathcal{S}_p}\right)^{p-2}}\mathrm{tr}\left(\mathbf{A}\cdot\mathbf{B}^*\left(|\mathbf{B}|_m\right)^{p-2}\right) \tag{3.38}$$

with $|\mathbf{B}|_m = \sqrt{\mathbf{B}\mathbf{B}^*}$ or, equivalently,

$$[\mathbf{A},\mathbf{B}]_{\mathcal{S}_p} = \frac{1}{\left(\|\mathbf{B}\|_{\mathcal{S}_p}\right)^{p-2}}\mathrm{tr}\left(\mathbf{A}\cdot\mathbf{B}^*\left(|\mathbf{B}|_n\right)^{p-2}\right) \tag{3.39}$$

is valid with $|\mathbf{B}|_n = \sqrt{\mathbf{B}^*\mathbf{B}}$. *Thus it fulfills the respective Cauchy-Schwarz-inequality*

$$\left|[\mathbf{A},\mathbf{B}]_{\mathcal{S}_p}\right|^2 \leq [\mathbf{A},\mathbf{A}]_{\mathcal{S}_p}[\mathbf{B},\mathbf{B}]_{\mathcal{S}_p} \tag{3.40}$$

as required for a SIP.

The proof of this theorem is given in Appendix (B.3).

Note 3.16. Further, it turns out that the real SIP $[\mathbf{A},\mathbf{B}]^*_{\mathcal{S}_p} : \mathcal{B}^*_{m,n}\times\mathcal{B}^*_{m,n}\to\mathbb{R}$ for Banach spaces $\mathcal{B}^*_{m,n} = \mathbb{R}^{m\times n}$ is also linear in the second argument and, hence, it generates a linear operator

$$\mathscr{F}_{\mathbf{A}}[\mathbf{B}] = [\mathbf{A},\mathbf{B}]^*_{\mathcal{S}_p}\cdot\mathbf{A} \tag{3.41}$$

in $\mathcal{B}^*_{m,n}$ according to [83].

3.6.3 QR-Norms

G. Allen introduced in [4] the **QR**-norm

$$\|\mathbf{A}\|_{\mathbf{Q},\mathbf{R}} = \sqrt{\mathrm{tr}\,(\mathbf{QARA}^*)} \tag{3.42}$$

with positive (semi-)definite matrices $\mathbf{Q}$ and $\mathbf{R}$ without any proof for this proposition. $\|\mathbf{A}\|_{\mathbf{Q},\mathbf{R}}$ can be seen as an extension of Schatten-p-norms, because for $\mathbf{Q} = \mathbf{I}$ and $\mathbf{R} = \mathbf{I}$ the **QR**-norm reduces to the Schatten-p-norm. The corresponding distance $d_{\mathbf{Q},\mathbf{R}}(\mathbf{A},\mathbf{B}) = \|\mathbf{A}-\mathbf{B}\|_{\mathbf{Q},\mathbf{R}}$ has been used by G. Allen to identify fMRI-voxel-time-series for appropriately chosen positive (semi-) definite matrices $\mathbf{Q}$ and $\mathbf{R}$ to scale adequately the spatial and the time resolution of the fMRI-series.

Lemma 3.17. *The norm $\|\mathbf{A}\|_{\mathbf{Q},\mathbf{R}}$ from (3.42) constitutes a vector norm for matrices for positive definite symmetric matrices $\mathbf{Q}$ and $\mathbf{R}$.*

The proof is given in Appendix B.5. Note that, $\|\mathbf{QAR}\|_{S_2}$ is not necessarily a matrix norm although the Frobenius $\|\mathbf{A}\|_{S_2}$ is a matrix norm [50, p. 371, remark after example 1].

Remark 3.18. The proof implies that $\mathbf{Q}$ and $\mathbf{R}$ both have to be symmetric matrices for a valid norm $\|\mathbf{A}\|_{\mathbf{Q},\mathbf{R}}$. Thus, the used decomposition $\mathbf{Q} = \tilde{\mathbf{Q}}^*\tilde{\mathbf{Q}}$ and $\mathbf{R} = \tilde{\mathbf{R}}\tilde{\mathbf{R}}^*$, see Appendix B.5 equation (B.12), is the Cholesky decomposition of the positive symmetric matrices $\mathbf{Q}$ and $\mathbf{R}$.

Further, the norm $\|\mathbf{A}\|_{\mathbf{Q},\mathbf{R}}$ is in general not unitarily invariant. More precisely, the following lemma is valid:

Lemma 3.19. *The norm $\|\mathbf{A}\|_{\mathbf{Q},\mathbf{R}}$ is unitarily invariant iff either $\mathbf{R} = \mathbf{I}$ or $\mathbf{Q} = \mathbf{I}$ is valid.*

The proof is stated in Appendix B.6. The question arises, whether the norm is generated by an inner product or by a semi-inner product. It is answered by the next lemma:

Lemma 3.20. *The norm $\|\mathbf{A}\|_{\mathbf{Q},\mathbf{R}}$ is generated by an inner product defined by*

$$\langle \mathbf{A},\mathbf{B}\rangle_{\mathbf{Q},\mathbf{R}} = \mathrm{tr}\,(\mathbf{QARB}^*) \tag{3.43}$$

for positive definite symmetric matrices $\mathbf{Q}$ and $\mathbf{R}$, i. e. also the Hermitian symmetries $\mathbf{Q} = \mathbf{Q}^*$ and $\mathbf{R} = \mathbf{R}^*$ are valid.

The proof is given in Appendix B.7. Note that, $\langle \mathbf{A},\mathbf{B}\rangle_{\mathbf{Q},\mathbf{R}}$ is sesquilinear due to the

Hermitian symmetry and the linearity in the first argument, i. e. $\langle \mathbf{A}, \mathbf{B} \rangle_{\mathbf{Q},\mathbf{R}}$ is a sesquilinear form [136].

Chapter 4

Hebbian Learning Based on General Inner and Semi-inner Products

Principal Component Analysis (PCA) and Independent Component Analysis (ICA) based on Hebbian learning were originally developed for data processing in Euclidean spaces, as mentioned in the introduction. A variety of nonlinear extensions of PCA and ICA were suggested in the literature. In detail, Kernel Hebbian learning was proposed by Kim et al. in [70, 69]. This approach is based on the general idea of kernel PCA using reproducing kernel Hilbert spaces (RKHS) [49, 122], which offer the possibility to capture non-linear data structures while applying PCA. An improvement of this approach was suggested by S. Günther et al. in [39], where an accelerating gain parameter was introduced. Hebbian PCA learning for functional data by using a special case of the Sobolev inner product was proposed in [139].
Further, nonlinear Hebbian approaches of ICA are generalizations of the linear approaches, see [55, 63]. Several kernel methods for ICA are suggested in [8, 42, 90], which can separate nonlinear mixed sources. But these procedures are no type of Hebbian-like learning.

This chapter will address unified and generalized Hebbian approaches for PCA and ICA in non-Euclidean spaces. It is shown that Hebbian learning for PCA can be performed in Banach spaces using the underlying semi-inner product (SIP). In detail l_p-spaces for $p \neq 2$, Sobolev- and kernel spaces are considered. For kernel spaces

the Hebbian approach can be formulated as an online learning scheme based on differentiable kernels. However, explicit handling of data in this non-Euclidean kernel mapping space is not possible. Therefore, an isomorphic mapping space, as introduced in subsection 3.5 on page 46, is employed and provides an alternative. The same problem is valid by considering Hebbian learning for nonlinear ICA by means of kernels. In that case it can proceed as before. Hence, PCA and ICA can be performed in the concerning data spaces but are equipped with a non-Euclidean metric.

The previously mentioned Hebbian PCA approaches process vectorial data, however, analysis of image data is frequently necessary. If the analysis of image data is based on matrix norms or distance measures the calculation of them is usually expensive due to the matrix size. Therefore, a low-dimensional feature representation becomes desirable. Mostly, an explicit feature extraction based on image processing tools is used or the data matrices treated as vectors equipped with l_p-norms and a subsequent PCA for both approaches [83, 14, 125]. However, both methods are associated with an information loss [17]. For this reason, Hebbian PCA learning for matrices is considered by using Schatten-p-norms as an alternative. The mathematical theory of *eigenmatrices* defined for Banach spaces of matrices and respective principal components will be introduced. This requires the SIP (semi-inner product) of Schatten-p-norms, which is a result of chapter 3. The presented algorithm of Hebbian PCA Learning for matrices determines the principal components of a given matrix dataset, which can be used for complexity reduction. Further, in the next chapter the properties of matrix approaches are investigated in deeper detail in applications like image classification.

This chapter is structured as follows: First Euclidean Hebbian PCA Learning is extended to general finite dimensional Hilbert-spaces, which are isomorphic to the Euclidean space (the original Oja learning rule). Thereafter, this idea is transferred to Hebbian PCA learning in Banach spaces including the concept of semi-inner products. Next, this method is further extended to kernel spaces. After that, the Hebbian-like algorithm for nonlinear ICA in general Reproducing Kernel spaces is presented to show that the theoretical considerations of Non-Euclidean PCA can be transferred to ICA. Further, vectorial Hebbian PCA learning is generalized to process matrix data. At the end of this chapter, example applications and different data sets are used to illustrated the new proposed Hebbian approaches and to demonstrate their usefulness.

The following subsections are based on

M. Lange, M. Biehl, T. Villmann, "Non-Euclidean Principal Component Analysis by Hebbian Learning", *Neurocomputing, 2015.[83]*

M. Biehl, M. Kästner, M. Lange, T. Villmann, "Non-Euclidean Principal Component Analysis and Oja's Learning Rule - Theoretical Aspects", *in P.A. Estevez, J.C. Principe, P. Zegers, ed., Advances in Self-Organizing Maps: 9th International Workshop WSOM 2012 Santiago de Chile vol. 198, (Berlin: Springer, 2013), pp. 23-34.[14]*

4.1 Hebbian Learning of PCA in Finite-dimensional Vector Spaces

This section deals with Hebbian learning for PCA in finite-dimensional Euclidean Hilbert and Banach spaces. In general, it is known that any kind of normalization influences PCA in Euclidean spaces. The same applies also for general Hilbert and Banach spaces. This should be merely recognized.

4.1.1 Hebbian PCA Learning in General Hilbert Space and Oja Learning

Let $\mathbf{v} = (v_1, ..., v_n)^\top$ be centered data in an n-dimensional Hilbert space $\mathcal{H}^n$ with the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}^n}$ generating the norm $\|\cdot\|_{\mathcal{H}^n}$. There is always an isomorphism $\Upsilon : \mathbb{R}^n \to \mathcal{H}^n$ because each n-dimensional Hilbert space $\mathcal{H}^n$ is isomorphic to the Euclidean space $\mathbb{R}^n$. Further, each linear operator constitutes a matrix $\mathbf{A}$ and the application of such an operator to a vector is defined by

$$\boldsymbol{A}[\mathbf{v}] = \left(\langle \mathbf{a}_1, \mathbf{v} \rangle_{\mathcal{H}^n}, \ldots, \langle \mathbf{a}_n, \mathbf{v} \rangle_{\mathcal{H}^n}\right)^\top \tag{4.1}$$

where the $\mathbf{a}_i$ are the row vectors of $\mathbf{A}$. With regard to *Oja's learning rule*

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta \left(O\mathbf{v}(t) - O^2 \mathbf{w}(t) \right)$$

the Euclidean inner product in the Hebb response $O = \langle \mathbf{v}, \mathbf{w} \rangle_E$ can be replaced by the inner product $O_{\mathcal{H}^n} = \langle \mathbf{v}, \mathbf{w} \rangle_{\mathcal{H}^n}$ of the Hilbert space yielding the learning rule

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta \left(\mathscr{F}_{\mathbf{v}}[\mathbf{w}] - \left(O_{\mathcal{H}^n}\right)^2 \mathbf{w}(t) \right) \tag{4.2}$$

where $\mathscr{F}_{\mathbf{v}}[\mathbf{w}]$ with

$$\mathscr{F}_{\mathbf{v}}[\mathbf{w}] = O_{\mathcal{H}^n} \cdot \mathbf{v}. \tag{4.3}$$

is a linear operator in the Hilbert space $\mathcal{H}^n$ due to the linearity of inner products.

In the next step the stationary state of (4.2) is investigated analogously to the Euclidean case. Under the assumption of a slowly changing $\mathbf{w}$ the eigenvalue equation

$$\mathbf{C}_{\mathcal{H}^n}[\mathbf{w}] = E\left[\left(O_{\mathcal{H}^n}\right)^2\right] \cdot \mathbf{w} \tag{4.4}$$

is obtained, where $\mathbf{C}_{\mathcal{H}^n}$ is the expectation of $\mathscr{F}_{\mathbf{v}}[\mathbf{w}]$ for all $\mathbf{v}$. More detailed, $\mathbf{C}_{\mathcal{H}^n}$ is the covariance operator (matrix) in $\mathcal{H}^n$ according to the basis representation of vectors in $\mathcal{H}^n$, i. e.

$$\begin{aligned} E[\mathscr{F}_{\mathbf{v}}[\mathbf{w}]] &= E[\mathbf{v} \cdot \langle \mathbf{v}, \mathbf{w} \rangle_{\mathcal{H}^n}] && (4.5)\\ &= E\left[\mathbf{v} \cdot \mathbf{v}^\top\right][\mathbf{w}] && (4.6)\\ &= \mathbf{C}_{\mathcal{H}^n}[\mathbf{w}], && (4.7) \end{aligned}$$

where the linearity of the inner product with respect to the first argument is used in the second step. The stability analysis of (4.4) shows that $\mathbf{w}$ converges to the eigenvector corresponding to the maximum eigenvalue of the covariance matrix $\mathbf{C}_{\mathcal{H}^n}$. This ensues immediately from the isomorphism $\Upsilon : \mathbb{R}^n \to \mathcal{H}^n$. The extension to Sanger's learning rule is straightforward.

Hebbian PCA Learning in Infinite but Separable Hilbert Spaces[1] The concept can be slightly adapted for infinite but separable Hilbert spaces $\mathcal{H}$. According to Zorn's-Lemma in [58] there is always a countable basis $\mathtt{H} = h_k \in \mathcal{H} | k \in \mathbb{N}$ with a respective unique representation $\mathbf{v} = \sum_{k=1}^{\infty} v_k h_k$ for all infinite-dimensional vectors $\mathbf{v} \in \mathcal{H}$. The covariance operator $\mathbf{C}_{\mathcal{H}}$ becomes infinite-dimensional but remains a linear operator. It is formally defined by the expectation $\mathbf{C}_{\mathcal{H}} = E\left[\mathbf{v} \cdot \mathbf{v}^\top\right]$ of infinite-dimensional vectors $\mathbf{v}$, which are represented according to a well-defined but infinite basis. The approximation property of PCA is ensured by the Riesz representer theorem and the Parseval's identity [109].

[1] If a Hilbert space $\mathcal{H}$ contains an uncountable orthonormal system, it cannot be separable. An example of non-separable Hilbert space is the following: Consider real valued functions and uses the inner product $\langle f, g \rangle = \lim_{R \to \infty} \frac{1}{R} \int_{-R}^{R} f(x)\, g(x)\, dx$.

4.1.2 Hebbian PCA Learning in Separable Banach Spaces

Each n-dimensional Banach space $\mathcal{B}^n$ is separable and countable with the finite basis $B = \{b_k \in \mathcal{B}^n\}$. Hence, there is a unique finite basis representation $\mathbf{v} = \sum_{k=1}^{\infty} v_k b_k$ for each vector $\mathbf{v}$. In n-dimensional Banach spaces $\mathcal{B}^n$ the application of a linear Operator $\mathbf{A}$ is defined with the SIP $[\cdot,\cdot]_{\mathcal{B}^n}$ as

$$\boldsymbol{A}[\mathbf{v}] = \left([\mathbf{a}_1, \mathbf{v}]_{\mathcal{B}^n}, \ldots, [\mathbf{a}_n, \mathbf{v}]_{\mathcal{B}^n}\right)^\top \tag{4.8}$$

analogously to (4.1).

For Hebbian PCA learning in Banach spaces let $\mathbf{v} \in \mathcal{B}^n$ be centered data, again. The Euclidean inner product in Oja's learning rule can be replaced by the SIP $[\cdot,\cdot]_{\mathcal{B}^n}$ and yields

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta\left(\mathscr{F}_{\mathbf{v}}[\mathbf{w}] - (O_{\mathcal{B}^n})^2 \mathbf{w}(t)\right) \tag{4.9}$$

where $O_{\mathcal{B}^n} = [\mathbf{v}, \mathbf{w}]_{\mathcal{B}^n}$ and, hence, $\mathscr{F}_{\mathbf{v}}[\mathbf{w}] = O_{\mathcal{B}^n} \cdot \mathbf{v}$. As before, a slowly changing $\mathbf{w}$ is assumed. The stationary state of the learning rule (4.9) corresponds to the eigenvalue equation

$$\mathbf{C}_{\mathcal{B}^n}[\mathbf{w}] = E\left[(O_{\mathcal{B}^n})^2\right] \cdot \mathbf{w} \tag{4.10}$$

where $\mathbf{C}_{\mathcal{B}^n}$ is the expectation of $\mathscr{F}_{\mathbf{v}}[\mathbf{w}]$ for all $\mathbf{v}$ and is still a linear operator. Completely analogously to separable Hilbert Spaces, $\mathbf{C}_{\mathcal{B}^n}$ can be seen as a generalized covariance operator (matrix) in $\mathcal{B}^n$ according to the given basis representation of vectors $\mathbf{v} \in \mathcal{B}^n$, i. e.

$$E[\mathscr{F}_{\mathbf{v}}[\mathbf{w}]] = E[\mathbf{v} \cdot [\mathbf{v}, \mathbf{w}]_{\mathcal{B}^n}] \tag{4.11}$$

$$= E\left[\mathbf{v} \cdot \mathbf{v}^\top\right][\mathbf{w}] \tag{4.12}$$

$$= \mathbf{C}_{\mathcal{B}^n}[\mathbf{w}], \tag{4.13}$$

where the linearity of the SIP in the first argument in the second line and $\mathbf{C}_{\mathcal{B}^n} = E\left[\mathbf{v} \cdot \mathbf{v}^\top\right]$ is used. [2]

The stability analysis of Oja's learning rule by E. Oja in [97, 98] takes only the norm properties into account. Thus, it is also applicable for learning rules including SIPs and, hence, in finite-dimensional Banach-spaces the updates converge to the eigenvector corresponding to the largest eigenvalue. As before, the extension to Sanger's learning rule is obvious.

[2] Note that, $\mathbf{C}_{\mathcal{B}^n}$ is called generalized covariance operator in $\mathcal{B}^n$ due to it is not necessarily symmetric. For a shorter notation $\mathbf{C}_{\mathcal{B}^n}$ is referred to as covariance operator in the following.

Hebbian PCA Learning in Infinite but Separable Banach Spaces In analogy to Hilbert spaces, the considerations are extended to Hebbian PCA Learning in infinite but separable Banach spaces. A (countable) Schauder basis B_s, which holds for reflexive Banach spaces [62] is supposed (see the explanations 3.13 on page 46). Further, the Schauder basis representation is unique and, hence, it can serve for approximated representations [60, 61, 107].

Furthermore, generalized SIPs, like described in subsection 3.1.1 on page 36, can also be applied in Hebbian PCA Learning. In that case the Hebb-response is generated by a generalized SIP. The respective Banach space also fulfill the additional constraints ensuring the separability and the existence of a (countable) Schauder basis.

The subsections are based on

M. Lange, M. Biehl, T. Villmann, "Non-Euclidean Principal Component Analysis by Hebbian Learning", *Neurocomputing, 2015.[83]*
M. Biehl, M. Kästner, M. Lange, T. Villmann, "Non-Euclidean Principal Component Analysis and Oja's Learning Rule - Theoretical Aspects", *in P.A. Estevez, J.C. Principe, P. Zegers, ed., Advances in Self-Organizing Maps: 9th International Workshop WSOM 2012 Santiago de Chile vol. 198, (Berlin: Springer, 2013), pp. 23-34.[14]*

4.2 Hebbian Learning for PCA in Reproducing Kernel Spaces

This subsection starts with a brief introduction of *Kernel Principal Component Analysis* (KPCA) and then the extension of Hebbian PCA Learning in Reproducing Kernel Spaces is presented.

4.2.1 Kernel PCA

Let the RKHS $\mathcal{H}$ be a mapping space and Φ a feature map of V with the corresponding kernel κ_Φ. Centralized kernels, i. e. $E\left[\Phi\left(\mathbf{v}\right)\right] = 0$, which can be always be accomplished for arbitrary positive definite kernels and finite data sets are assumed [121]. To perform PCA the covariance matrix is defined as $\mathbf{C}_\Phi = E\left[\Phi\left(\mathbf{v}\right) \cdot \left(\Phi\left(\mathbf{v}\right)\right)^\top\right]$. In infinite-dimensional $\mathcal{H}$ the term $\Phi\left(\mathbf{v}\right) \cdot \left(\Phi\left(\mathbf{v}\right)\right)^\top$ has to be interpreted as a linear operator $\Omega_\mathcal{H}$ on $\mathcal{H}$

$$\Omega_\mathcal{H}\left[\mathbf{h}\right] = \Phi\left(\mathbf{v}\right) \cdot \left\langle \Phi\left(\mathbf{v}\right), \mathbf{h} \right\rangle_\mathcal{H}. \tag{4.14}$$

According to Schölkopf Et Al. in [122] the respective eigenvalue equation

$$\mathbf{C}_\Phi \mathbf{g} = \lambda \mathbf{g}$$

can be solved by using the observation that the equation $\lambda \langle \Phi(\mathbf{v}), \mathbf{g} \rangle_{\mathcal{H}} = \lambda \langle \Phi(\mathbf{v}), \mathbf{C}_\Phi \mathbf{g} \rangle_{\mathcal{H}}$ has to be satisfied for all $\mathbf{v} \in V$. For a data set $D \subset V$ with m linear independent data vectors $\mathbf{v}_k$ there exits a dual representation of the eigenvectors $\mathbf{g} = \sum_{j=1}^{m} \alpha_j \Phi(\mathbf{v}_j)$. In this case $\mathbf{C}_\Phi$ becomes the *Gram-matrix* $\mathbf{G}_m$, see equation 3.25 on page 47. Thus the initial eigenvalue equation is the dual variant

$$m\lambda \boldsymbol{\alpha} = \mathbf{G}_m \boldsymbol{\alpha} \tag{4.15}$$

where $\boldsymbol{\alpha}$ is the column vector of the values α_i. Additionally according to Zhang et al. in [147], this eigenvalue equation can also be understood as an equation including a linear operator by using the kernel properties and reads as

$$\langle T\mathbf{c}, \mathbf{h} \rangle_{\mathcal{H}} = \frac{1}{m} \sum_{j=1}^{m} \langle \Phi(\mathbf{v}_j), \mathbf{c} \rangle_{\mathcal{H}} \langle \Phi(\mathbf{v}_j), \mathbf{h} \rangle_{\mathcal{H}}. \tag{4.16}$$

The approach of RKHS can be extended to RKBS [147]. For that as introduced in subsection 3.5 on page 48, considering RKBS $\mathcal{B}$ as a mapping space and a feature map Φ of V with the respective (centralized) SIP-kernel γ_Φ. Let $D \subset V$ be a data set with m linear independent data vectors $\mathbf{v}_k$. For an arbitrary $\mathbf{v} \in \mathcal{B}$ a complex m-dimensional vector

$$\tilde{\Phi}_{\mathcal{B}}(\mathbf{v}) = ([\Phi(\mathbf{v}), \Phi(\mathbf{v}_1)]_{\mathcal{B}}, \ldots, [\Phi(\mathbf{v}), \Phi(\mathbf{v}_m)]_{\mathcal{B}}) \tag{4.17}$$

can be defined such that a linear operator T is obtained

$$T\mathbf{c} = \frac{1}{m} \sum_{j=1}^{m} \left(\tilde{\Phi}^*_{\mathcal{B}}(\mathbf{v}_j)\, \mathbf{c} \right) \tilde{\Phi}_{\mathcal{B}}(\mathbf{v}_j). \tag{4.18}$$

$\tilde{\Phi}^*_{\mathcal{B}}(\mathbf{v}_j)$ is the conjugate transpose of $\tilde{\Phi}_{\mathcal{B}}(\mathbf{v}_j)$, which corresponds to $T\mathbf{c} = \mathbf{M}_m \mathbf{c}$ with

$$\mathbf{M}_m = \frac{1}{m} (\mathbf{K}^*_m \cdot \mathbf{K}_m)^\top \tag{4.19}$$

where

$$\mathbf{K}_m = [\gamma_\Phi(\mathbf{v}_i, \mathbf{v}_j) : i, j = 1 \ldots m] \tag{4.20}$$

is the Gram-matrix of the SIP-kernel γ_Φ. Therefore, the dual eigenvalue equation is

$$\mathbf{M}_m \boldsymbol{\alpha} = \lambda \boldsymbol{\alpha} \tag{4.21}$$

with the basis representation according to

$$\left\langle \tilde{\Phi}_{\mathcal{B}}(\mathbf{v}), \boldsymbol{\alpha} \right\rangle_{\mathbb{C}^m} = \sum_{j=1}^{m} \bar{\boldsymbol{\alpha}}_j \gamma_\Phi(\mathbf{v}_i, \mathbf{v}_j) \tag{4.22}$$

where $\bar{\boldsymbol{\alpha}}_j$ is the conjugate-complex of $\boldsymbol{\alpha}_j$.

4.2.2 Kernel PCA and Hebbian Learning in RKHS and RKBS

As mentioned above, an iterative method for performing KPCA is *Kernel Hebbian Algorithm* [70], which iteratively estimates the Kernel Principal Components in the Hilbert space $\mathcal{H}$ such that the coefficient vector $\boldsymbol{\alpha}$ in (4.15) is calculated by using the Gram-matrix $\mathbf{G}_m$. This procedure can be adapted to determine $\boldsymbol{\alpha}$ in a RKBS. For that, the terms containing $\mathbf{G}_m$ are replaced by the respective parts of $\mathbf{M}_m$. Note that, performing of Hebbian PCA learning in RKHS and RKBS uses implicit the mapping Ψ, which was introduced 3.5 on page 46.

Hebbian PCA Learning in $(V, d_{\mathcal{H}})$

Given is a data space V with metric d_V, which is most frequently the Euclidean metric d_E. Previously, PCA is considered in general Hilbert- and Banach spaces. Now, this method is transferred to the space $(V, d_{\mathcal{H}})$, see (3.28) on page 47. The isomorphism to the image space $\mathcal{I}_{\kappa_\phi} \subseteq \mathcal{H}$ of the kernel mapping Φ is used. Thus the original data is not changed, but additionally equipped with a kernel metric. The advantage is that the relations between the data objects are amended compared to the original data space (V, d_V) .

For Oja's learning rule again centralized kernels such that $E[\Psi(\mathbf{v})] = 0$ with $\mathbf{v} \in (V, d_V)$ are assumed. Thus, Oja's learning rule (2.10) in $(V, d_{\mathcal{H}})$ becomes

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta \left(O_{\mathcal{H}} \Psi(\mathbf{v}_k) - (O_{\mathcal{H}})^2 \mathbf{w}(t) \right), \tag{4.23}$$

where

$$O_{\mathcal{H}} = \kappa_\Phi(\Psi(\mathbf{v}_k), \mathbf{w}) \tag{4.24}$$

is now the (non-Euclidean) Hebb response. By means of substitution $O_{\mathcal{H}}$ in (4.23) results in

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta\left(\kappa_\Phi\left(\Psi\left(\mathbf{v}_k\right), \mathbf{w}\right)\Psi\left(\mathbf{v}_k\right) - \kappa_\Phi\left(\Psi\left(\mathbf{v}_k\right), \mathbf{w}\right)\kappa_\Phi\left(\Psi\left(\mathbf{v}_k\right), \mathbf{w}\right)\mathbf{w}(t)\right). \tag{4.25}$$

With $\Omega = \Psi\left(\mathbf{v}_k\right) \cdot \left(\Psi\left(\mathbf{v}_k\right)\right)^\top$ being a linear operator the learning rule (4.25) can be altered to

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta\left(\Omega\left[\mathbf{w}\right] - \kappa_\Phi\left(\Psi\left(\mathbf{v}_k\right), \mathbf{w}\right)\kappa_\Phi\left(\Psi\left(\mathbf{v}_k\right), \mathbf{w}\right)\mathbf{w}(t)\right). \tag{4.26}$$

Note that $\Omega = \Psi\left(\mathbf{v}_k\right) \cdot \left(\Psi\left(\mathbf{v}_k\right)\right)^\top$ is just a notation for the linear operator Ω in case of an infinite-dimensional Hilbert space $\mathcal{H}$. Comparable to the linear operator in (4.14) the operator equation with

$$\Omega\left[\mathbf{w}\right] = \Psi\left(\mathbf{v}_k\right) \cdot \kappa_\Phi\left(\Psi\left(\mathbf{v}_k\right), \mathbf{w}\right) \tag{4.27}$$

is valid. Note that at this point that $\Psi\left(\mathbf{v}_k\right) \in \mathcal{H}$ may, however, be infinite-dimensional vectors.

Considering the stationary state of (4.26) under the familiar assumption of a slowly changing $\mathbf{w}$ compared to the number of presented inputs the equation

$$\Delta\mathbf{w} = \mathbf{C}_\Psi\left[\mathbf{w}\right] - \lambda\mathbf{w} \tag{4.28}$$

results, where

$$\mathbf{C}_\Psi = \mathrm{E}\left[\Omega\right] \tag{4.29}$$

defines the covariance in $(V, d_{\mathcal{H}})$. For a finite number of data samples $D = \{\mathbf{v}_k \mid k = 1, \ldots, m\} \subseteq V$ the covariance reduces to

$$\mathbf{C}_\Psi = \frac{1}{m}\sum_{j=1}^{m} \Psi\left(\mathbf{v}_j\right) \cdot \left(\Psi\left(\mathbf{v}_j\right)\right)^\top. \tag{4.30}$$

The value λ in (4.15) is the expectation

$$\lambda = \mathrm{E}\left[\kappa_\Phi\left(\Psi\left(\mathbf{v}_k\right), \mathbf{w}\right) \cdot \kappa_\Phi\left(\Psi\left(\mathbf{v}_k\right), \mathbf{w}\right)\right] \tag{4.31}$$

of the squared (non-Euclidean) Hebbian response $O_{\mathcal{H}}$ in (4.24). The stationary state $\Delta\mathbf{w} = 0$ yields an eigenvalue equation $\mathbf{C}_\Psi\left[\mathbf{w}\right] = \lambda\mathbf{w}$ with the operator $\mathbf{C}_\Psi$ for an eigenvector $\mathbf{w} \neq 0$ and eigenvalues $\lambda > 0$. The positivity of the eigenvalues is ensured

by positive definiteness of the kernel.

Further, $\mathbf{w} \in \operatorname{span}\{\Psi(\mathbf{v}_j) \mid j = 1, \ldots, m\}$ is valid due to $\mathbf{w} \in (V, d_{\mathcal{H}})$ and, hence, the relation

$$\lambda \kappa_\Phi(\Psi(\mathbf{v}_k), \mathbf{w}) = \kappa_\Phi(\Psi(\mathbf{v}_k), \mathbf{C}_\Psi[\mathbf{w}]) \tag{4.32}$$

has to be satisfied for all $k = 1, \ldots, n$. In addition, $\mathbf{w}$ can be written as a linear combination

$$\mathbf{w} = \sum_{j=1}^{m} \alpha_j \Psi(\mathbf{v}_j) \tag{4.33}$$

consisting of the images $\Psi(\mathbf{v}_k)$ of the original data vectors. Taking this into account, the relation (4.32) changes to

$$\lambda \sum_{j=1}^{m} \alpha_j \kappa_\Phi(\Psi(\mathbf{v}_k), \Psi(\mathbf{v}_j)) = \frac{1}{m} \sum_{j=1}^{m} \alpha_j \kappa_\Phi\left(\Psi(\mathbf{v}_k), \sum_{i=1}^{m} \Psi(\mathbf{v}_i) \cdot \kappa_\Phi(\Psi(\mathbf{v}_i) \cdot \Psi(\mathbf{v}_j))\right). \tag{4.34}$$

In (4.34) the definition of $\mathbf{C}_\Psi$ in (4.29) is used as well as the linearity of the kernel, which can be interpreted as a real inner product. If now the definition of the Gram-matrix $\mathbf{G}_m = [\kappa_\phi(\mathbf{v}_i, \mathbf{v}_j) : i, j = 1 \ldots m]$ is used in (4.34), the relation

$$m\lambda \mathbf{G}_m \boldsymbol{\alpha} = \mathbf{G}_m^2 \boldsymbol{\alpha} \tag{4.35}$$

results, where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)^\top$. The equation (4.35) corresponds to the solution of the so called dual variant of the eigenvalue equation (4.15) in [121]. Hence, the stability analysis of the dynamic (4.23) can be taken from [70], which also provides the extension to the full problem of the eigenvalue equation and additionally the respective Sanger's learning rule.

Hebbian PCA Learning in $(V, d_{\mathcal{B}})$

Hebbian PCA learning in the space $(V, d_{\mathcal{B}})$ use the isomorphism to subspace $\mathcal{I}_{\gamma_\phi} \subseteq \mathcal{B}$ of the kernel mapping Φ for a SIP-RKBS $\mathcal{B}$. The considered RKBS space $\mathcal{B}$ is reflexive. According to remark 3.13 on page 46 a (countable) Schauder basis can always be supposed for reflexive Banach spaces.

Again, centralized kernels satisfying $E[\Psi(\mathbf{v})] = 0$ are assumed and additionally the kernel γ_Φ has merely real values. Thus, $\mathbf{K}_m^* = \mathbf{K}_m^\top$ is valid and (4.19) becomes $\mathbf{M}_m = \frac{1}{m}\left(\mathbf{K}_m^\top \cdot \mathbf{K}_m\right)^\top$, which is symmetric and positive definite. Oja's learning rule

in $(V, d_{\mathcal{B}})$ using the respective (non-Euclidean) Hebb response

$$O_{\mathcal{B}} = \gamma_{\Phi}\left(\Psi\left(\mathbf{v}_k\right), \mathbf{w}\right) \tag{4.36}$$

reads here as

$$\mathbf{w}\left(t+1\right) = \mathbf{w}\left(t\right) + \eta\left(\gamma_{\Phi}\left(\Psi\left(\mathbf{v}_k\right), \mathbf{w}\right)\Psi\left(\mathbf{v}_k\right) - \gamma_{\Phi}\left(\Psi\left(\mathbf{v}_k\right), \mathbf{w}\right)\gamma_{\Phi}\left(\Psi\left(\mathbf{v}_k\right), \mathbf{w}\right)\mathbf{w}\right). \tag{4.37}$$

In complete analogy to the case $(V, d_{\mathcal{H}})$, the equation

$$\Delta\mathbf{w} = \mathrm{E}\left[\Omega_{\mathcal{B}}\left[\mathbf{w}\right]\right] - \lambda\mathbf{w} \tag{4.38}$$

is valid for $(V, d_{\mathcal{B}})$, where $\Omega_{\mathcal{B}}\left[\mathbf{w}\right] = \Psi\left(\mathbf{v}_k\right) \cdot \gamma_{\Phi}\left(\Psi\left(\mathbf{v}_k\right), \mathbf{w}\right)$ is a linear operator and $\mathbf{C}_{\Psi}^{\mathcal{B}} = \mathrm{E}\left[\Omega_B\right]$. [3] In (4.38) the value λ represents the expectation

$$\lambda = \mathrm{E}\left[\gamma_{\Phi}\left(\Psi\left(\mathbf{v}_k\right), \mathbf{w}\right) \cdot \gamma_{\Phi}\left(\Psi\left(\mathbf{v}_k\right), \mathbf{w}\right)\right] \tag{4.39}$$

of the squared (non-Euclidean) Hebb response $O_{\mathcal{B}}$. $\mathbf{C}_{\Psi}^{\mathcal{B}} = \mathrm{E}\left[\Psi\left(\mathbf{v}_k\right) \cdot \Psi\left(\mathbf{v}_k\right)^{\top}\right]$ can be interpreted as the covariance operator (matrix) according to the basis representation of vectors in $\mathcal{B}$, i. e.

$$\begin{aligned}
\mathrm{E}\left[\Omega_B\left[\mathbf{w}\right]\right] &= \mathrm{E}\left[\Psi\left(\mathbf{v}_k\right) \cdot \gamma_{\Phi}\left(\Psi\left(\mathbf{v}_k\right), \mathbf{w}\right)\right] && (4.40)\\
&= \mathrm{E}\left[\Psi\left(\mathbf{v}_k\right) \cdot \Psi\left(\mathbf{v}_k\right)^{\top}\right]\left[\mathbf{w}\right] && (4.41)\\
&= \mathbf{C}_{\Psi}^{\mathcal{B}}\left[\mathbf{w}\right]. && (4.42)
\end{aligned}$$

The stationary state $\Delta\mathbf{w} = 0$ corresponds to an eigenvalue equation $\mathbf{C}_{\Psi}^{\mathcal{B}}\left[\mathbf{w}\right] = \lambda\mathbf{w}$ with eigenvector $\mathbf{w} \neq 0$ and eigenvalue $\lambda \neq 0$.

Let $\mathbf{v}_j \in V$, $j = 1, \ldots, m$ be data vectors. From $\mathbf{w} \in (V, d_{\mathcal{B}})$ follows $\mathbf{w} \in \mathrm{span}\left\{\Psi\left(\mathbf{v}_j\right) \mid j = 1, \ldots, m\right\}$, because $\mathcal{B}$ is a SIP-RKBS. Thus, for all $k = 1, \ldots, m$ the relation

$$\lambda\gamma_{\Phi}\left(\Psi\left(\mathbf{v}_k\right), \mathbf{w}\right) = \gamma_{\Phi}\left(\Psi\left(\mathbf{v}_k\right), \mathbf{C}_{\Psi}^{\mathcal{B}}\left[\mathbf{w}\right]\right) \tag{4.43}$$

is valid and $\mathbf{w}$ can be expressed again as linear combination $\mathbf{w} = \sum_{j=1}^{m} \alpha_j \Psi\left(\mathbf{v}_j\right)$ consisting of the images $\Psi\left(\mathbf{v}_k\right)$ of the original data vectors. The combination of the

[3] At this point it should be noticed that, $\Psi\left(\mathbf{v}_k\right) = \mathbf{v}_k$ holds only numerically. Yet, $\mathbf{v}_k$ and its image $\Psi\left(\mathbf{v}_k\right)$ are objects in different metric spaces. Therefore, the notation $\Psi\left(\mathbf{v}_k\right)$ for the image is still used to indicate this difference.

latter statement with (4.43) leads to

$$\lambda \sum_{j=1}^{m} \beta_j \gamma_\Phi \left(\Psi \left(\mathbf{v}_k \right), \Psi \left(\mathbf{v}_j \right) \right) = \frac{1}{m} \sum_{j=1}^{m} \beta_j \gamma_\Phi \left(\Psi \left(\mathbf{v}_k \right), \sum_{i=1}^{m} \Psi \left(\mathbf{v}_i \right) \cdot \gamma_\Phi \left(\Psi \left(\mathbf{v}_i \right) \cdot \Psi \left(\mathbf{v}_j \right) \right) \right), \tag{4.44}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_m)^\top$ has the same functionality as $\boldsymbol{\alpha}$ in (4.35). In (4.44) the linearity of the SIP-kernel in its first argument is used. γ_Φ can be seen as a real semi-inner product and the definition of $\mathbf{C}_\Psi^{\mathcal{B}}$ as expectation. If the definition of the Gram-matrix $\mathbf{K}_m$ in (4.20) is taken into account

$$m\lambda \mathbf{K}_m \alpha = \mathbf{K}_m^2 \alpha$$

results, which relates to the dual problem in case of RKBS in (4.21) due to the respective operator of the eigenvalue equation for RKHS (4.16) and RKBS (4.18).

The stability analysis of (4.37)for RKBS can be derived analogously as in [70], as explained in [97], because the sesquilinearity of the SIP is not used for this investigation, but only the resulting norm is taken into account. Again, the extension to the full problem of the eigenvalue equation for PCA and the respective Sanger's learning rule [115] is straightforward.

The subsection is based on

M. Lange, M. Biehl, T. Villmann, "Non-Euclidean Independent Component Analysis and Oja's Learning", *in M. Verleysen, ed., Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2013) (Louvain-La-Neuve, Belgium: i6doc.com, 2013), pp. 125-130.[80]*

4.3 Hebbian Learning of ICA in Reproducing Kernel Spaces

The conventional ICA constitutes a procedure to extract independent sources from a sequence of mixtures as introduced in subsection 2.4 on page 17. The application of kernels mappings enable nonlinear but implicit data mapping of the data into a RKHS, where a linear demixing can be executed afterwards, i. e. kernel ICA uses a nonlinear mapping to perform a nonlinear ICA $\mathbf{v}(t) = h(\mathbf{s}(t))$ for an unknown nonlinear function h. As indicated at the beginning of chapter 4, the known nonlinear kernel ICA approaches are not a kind of Hebbian like learning scheme. In the section before, Hebbian Kernel PCA learning for RKHS and RKBS was investigated such that

PCA can be executed using kernels. This idea will now be transferred to perform a non-Euclidean ICA. The following considerations are based on the approach presented by HARMELING ET AL. in [42].

Nonlinear Kernel ICA by Hebbian Learning in $(V, d_{\mathcal{H}})$

Let $\mathbf{v} \in (V, d_{\mathbf{V}})$ be an n-dimensional mixture vector. As before $\Phi : V \ni \mathbf{v} \to \Phi(\mathbf{v}) \in \mathcal{H}$ is a general nonlinear kernel data map with a positive definite kernel $\kappa_\phi(\mathbf{v}, \mathbf{w}) = \langle \Phi(\mathbf{v}), \Phi(\mathbf{w}) \rangle_{\mathcal{H}}$ and $\mathcal{I}_{\kappa_\Phi}$ is the image of the subspace of $\mathcal{H}$. If b_i forms an orthonormal basis in $\mathcal{I}_{\kappa_\Phi}$, then

$$\Phi(\mathbf{v}) = \sum_i \langle \Phi(\mathbf{v}), \mathrm{b}_i \rangle_{\mathcal{H}} \cdot \mathrm{b}_i \tag{4.45}$$

is the representation of an image vector $\Phi(\mathbf{v})$ in $\mathcal{I}_{\kappa_\Phi}$. The linear mixing problem in the Hilbert space $\mathcal{H}$ is defined as

$$\Phi(\mathbf{v}) = \mathbf{M}_{\mathcal{H}}[\Phi(\mathbf{s})], \tag{4.46}$$

where $\mathbf{M}_{\mathcal{H}}$ is a linear operator in $\mathcal{H}$. $\mathbf{M}^k_{\mathcal{H}}$ denotes the kth component of $\mathbf{M}_{\mathcal{H}}$. Each linear operator can be formulated in terms of inner products, i. e.

$$\mathbf{M}^k_{\mathcal{H}}[\Phi(\mathbf{s})] = \left\langle \mathbf{M}^k_{\mathcal{H}}, \Phi(\mathbf{s}) \right\rangle_{\mathcal{H}}, \tag{4.47}$$

because $\mathcal{H}$ is a RKHS. Including the basis representation (4.45),

$$\begin{aligned} \Phi_k(\mathbf{v}) &= \left\langle \mathbf{M}^k_{\mathcal{H}}, \sum_i \langle \Phi(\mathbf{s}), \mathrm{b}_i \rangle_{\mathcal{H}} \cdot \mathrm{b}_i \right\rangle_{\mathcal{H}} \\ &= \sum_i \langle \Phi(\mathbf{s}), \mathrm{b}_i \rangle_{\mathcal{H}} \cdot \left\langle \mathbf{M}^k_{\mathcal{H}}, \mathrm{b}_i \right\rangle_{\mathcal{H}} \end{aligned} \tag{4.48}$$

is valid, where $\langle \Phi(\mathbf{s}), \mathrm{b}_i \rangle$ is a random quantity due to the stochastic nature of $\mathbf{s}$. Consequently, $\Phi(\mathbf{s})$ is random too. According to CLT it can be stated that the sum in (4.48) is more Gaussian than the single components. The absolute kurtosis of $\Phi_k(\mathbf{v})$ has to be maximized to identify independent components in $\mathcal{I}_{\kappa_\phi}$. This is due to the same arguments as for linear ICA, see subsection 2.4 on page 17.

For a finite number N of samples in V the number of basis elements b_i is referred to

as $D \leq \min(n, N)$ and, thus, equation (4.48) can simplified to

$$\Phi_k(\mathbf{v}) = D \cdot \kappa_\Phi\left(\Phi^{-1}\left(\mathbf{M}_{\mathcal{H}}^k\right), \mathbf{s}\right). \tag{4.49}$$

In (4.49) the reproducing property of the inner product of a RKHS and the definition κ_Φ from (3.24) get also involved. Analogous to Kernel PCA by Hebbian learning, the kernel map Φ is now replaced by the mapping $\Psi : (V, d_V) \to (V, d_{\kappa_\Phi})$ with $d_{\kappa_\phi}(\mathbf{v}, \mathbf{w}) = d_{\mathcal{H}}(\Phi(\mathbf{v}), \Phi(\mathbf{w}))$. Formally, Ψ is the identity map from the one to the other metric space and, hence, the metric is changed. Thus, Ψ is nonlinear in general.

Further, the operator $\mathbf{M}_{\mathcal{H}}$ is equivalent to a conventional matrix $\mathbf{M}$ because the kernel space (V, d_{κ_Φ}) is isometric and isomorphic to $\mathcal{I}_{\kappa_\phi}$. However $\mathbf{M}[\Psi(\mathbf{s})]$ is defined by

$$\mathbf{M}[\Psi(\mathbf{s})] = D \cdot \kappa_\Phi(\mathbf{m}_k, \mathbf{s}),$$

where $\mathbf{m}_k$ is the kth row vector of $\mathbf{M}$.

A Non-Euclidean Two-Unit-Learning-Rule (TULR) for Whitened Data

Hebbian ICA learning in $(V, d_{\mathcal{H}})$ assumes a sequence of whitened input vectors $\mathbf{v}$ in the related space, i. e. if the pre whitening is done in $(V, d_{\mathcal{H}})$ then the correlation operator $\mathbf{C}_\Psi$ defines the covariance in $(V, d_{\mathcal{H}})$, hence, rotation (change of coordinate axes) in $(V, d_{\mathbf{V}})$ preserves kernel metric. An inconsistent use of norms would make the ICA algorithm very unstable. Note that, centralized kernels must also be applied here, such that $E[\Psi(\mathbf{v})] = 0$ is valid. A pre whitening of $\mathbf{v}$ in $(V, d_{\mathcal{H}})$ can be performed with Hebbian PCA Learning in $(V, d_{\mathcal{H}})$, such that the original data are decorrelated with the respective kernel metric. Therefore, Oja's learning rule (4.23) on page 62 constitutes an appropriate preprocessing step for Oja ICA learning in $(V, d_{\mathcal{H}})$.

The Non-Euclidean ICA approach is based on the learning rule (2.4.2). A non-Euclidean two units learning rule for whitened data in $(V, d_{\mathcal{H}})$ is defined as

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \mu(t)\left[\sigma \mathbf{v}(t)\,\kappa_\Phi(\Psi(\mathbf{v}_k), \mathbf{w})^3 - \left(\kappa_\Phi(\mathbf{w}, \mathbf{w})^2 - 1\right)\mathbf{w}(t)\right], \tag{4.50}$$

where $\sigma = \operatorname{sign}\left(\widehat{\operatorname{kurt}}(t)\right)$ is the sign of the kurtosis again. More precisely, the Euclidean inner products $\langle \mathbf{v}, \mathbf{w}\rangle_E$ and $\langle \mathbf{w}, \mathbf{w}\rangle_E$ of the original approach are replaced by the kernels $\kappa_\Phi(\Psi(\mathbf{v}_k), \mathbf{w})$ and $\kappa_\Phi(\mathbf{w}, \mathbf{w})$, respectively. Note that now the inner

products in $\widehat{\text{kurt}}\,(t)$ have to be replaced by kernels such that the estimation

$$\widehat{\text{kurt}}\,(t) = \widehat{m^4}\,(t) - 3\kappa_\Phi\left(\mathbf{w},\mathbf{w}\right)^2 \tag{4.51}$$

with

$$\widehat{m^4}\,(t+1) = (1-\nu)\,\widehat{m^4}\,(t) + \nu\kappa_\Phi\left(\Psi\left(\mathbf{v}_k\right),\mathbf{w}\right)^4 \tag{4.52}$$

is determined. The obtained Hebbian-like kernel ICA in $(V, d_{\mathcal{H}})$ is based on the original data and realizes a non-linear separation due to the non-linear kernel mapping Φ or its analogon Ψ.

Further, the stability analysis draws upon the Euclidean case in [54] by Hyvärinen & Oja. All independent components in $(V, d_{\mathcal{H}})$ can be estimated by using an extended variant of the the learning rule (4.50) with several units. Note that, analog to Hebbian PCA learning in RKBS, where the mapping Ψ is used implicitly, can also be defined Hebbian ICA learning in RKBS in a straightforward manner.

The subsection is based on

*M. Lange, D. Nebel and T. Villmann, "*Non-Euclidean Principal Component Analysis for Matrices by Hebbian Learning*", in L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh and J.M. Zurada, ed., Artificial Intelligence and Soft Computing - Proc. the International Conference ICAISC vol. 8467, (Zakopane: Springer, 2014), pp. 77-88.[81]*

*A. Villmann, M. Lange-Geisler, T. Villmann, "*About Semi-Inner Products for $\mathbf{p}-\mathbf{QR}$-Matrix Norms*", TechReport (2018).[136]*

4.4 Hebbian PCA Learning for Matrices

The last extended Hebbian PCA approach in this chapter provide a learning scheme based on Schatten-p-norms in the respective Banach space of matrices, i. e. the original Oja Sanger algorithm is generalized to a matrix approach, which enables the extraction principal components for given matrix data. For that *eigenmatrices* with respect to a covariance operator defined for datasets of matrices and respective principal components are defined as well as the convergence of the respective learning rule for matrices is proven [81].

4.4.1 Principal Components in $\mathcal{B}_{m,n}$

Let $B\left(n \cdot m\right) = \{\mathbf{b}_k\}_{k=1,\ldots,n\cdot m}$ be a basis in the vector space $\mathcal{B}_{m,n}$. A linear operator $\mathbf{P}_{m,n}^{l,p} : \mathcal{B}_{m,n} \to \mathbb{C}^l$ is defined by

$$\mathbf{P}_{B(n\cdot m)}^{l,p}\left[\mathbf{A}\right] = \left(\left[\mathbf{A}, \mathbf{b}_1\right]_{\mathcal{S}_p}, \ldots, \left[\mathbf{A}, \mathbf{b}_l\right]_{\mathcal{S}_p}\right)^\top, \tag{4.53}$$

where $1 \leq l \leq n \cdot m$ and $[\cdot,\cdot]_{\mathcal{S}_p}$ is the semi inner product (SIP) of the Schatten-p-norm. Further, a set $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots \mid \mathbf{s}_k \in \mathcal{B}_{1,m\cdot n}\}$ is a assumed. The linear generalized covariance operator $\mathbf{C}_\mathcal{S} \in \mathcal{B}^{[m\cdot n]\times[m\cdot n]}$ can be defined by the expectation $\mathbf{C}_\mathcal{S} = \mathrm{E}\left[\mathcal{S} \cdot \mathcal{S}^*\right]$. A matrix $\mathbf{X} \in \mathcal{B}_{m,n}$ with $\mathbf{X} \neq \mathbf{0}$ is called an *eigenmatrix* of the operator $\mathbf{C}_\mathcal{S}$ if the (vectorized) eigenmatrix equation

$$\mathbf{C}_\mathcal{S}\left[\mathbf{x}\right] = \lambda\mathbf{x} \tag{4.54}$$

is valid, where $\mathbf{x} \in \mathcal{B}_{m\cdot n,1}$ is the vectorization of $\mathbf{X}$. The scalar λ is the eigenvalue assigned to $\mathbf{X}$. The vector $\boldsymbol{\lambda} = \{\lambda_k \mid k = 1, \ldots, n\cdot m\}$ of all eigenvalues forms a spectrum of $\mathbf{C}_\mathcal{S}$. If $\mathbf{C}_\mathcal{S}$ is a regular operator, i. e. $\mathbf{C}_\mathcal{S}^{-1}$ exists, then the respective eigenmatrices $\mathbf{X}_k$ generates a basis in $\mathcal{B}_{m,n}$ because $\mathcal{B}_{m,n}$ is a vector space itself. Note that, the eigenmatrices of $\mathbf{C}_\mathcal{S}$ are the principal components.

4.4.2 Hebbian learning of Principal Components in $\mathcal{B}_{m,n}$

The principal components in $\mathcal{B}_{m,n}^* = \mathbb{R}^{m\times n}$ can be obtained with an extended version of Oja's learning rule. Let $\mathcal{V} = \left\{\mathbf{V}_k \mid k = 1, \ldots, N,\ \mathbf{V}_k \in \mathcal{B}_{m,n}^*\right\}$ be set of centered matrices and $W = \left\{\mathbf{W}_k \mid k = 1, \ldots, K,\ \mathbf{W}_k \in \mathcal{B}_{m,n}^*\right\}$ a randomly initialized set with $K = \min\left(n \cdot m, N\right)$. For randomly chosen matrices $\mathbf{V}_l \in \mathcal{V}$ and a learning rate $0 < \eta \ll 1$ the principal components can be generated by the learning rule

$$\Delta\mathbf{W}_k = \eta \cdot \left[\mathbf{V}_l, \mathbf{W}_k\right]_{\mathcal{S}_p}^* \cdot \left(\mathbf{V}_l - \sum_{j=1}^{k} \left[\mathbf{V}_l, \mathbf{W}_j\right]_{\mathcal{S}_p}^* \cdot \mathbf{W}_j\right), \tag{4.55}$$

where $[\cdot,\cdot]_{\mathcal{S}_p}^*$ denotes the real SIP $\left[\mathbf{A}, \mathbf{B}\right]_{\mathcal{S}_p}^* : \mathcal{B}_{m,n}^* \times \mathcal{B}_{m,n}^* \to \mathbb{R}$. The following lemma can be stated:

Lemma 4.1. *The learning rule (4.55) converges such that the matrices $\mathbf{W}_k$ are the eigenmatrices according to the eigenmatrix equation (4.54) corresponding to the K largest eigenvalues of the covariance operator $\mathbf{C}_V$ of the dataset $\mathcal{V}$.*

Proof. Since $\mathcal{B}_{m,n}$ is a Banach space with the SIP $[\cdot,\cdot]_{\mathcal{S}_p}$ generating the norm $\|\cdot\|_{\mathcal{S}_p}$, the follows by the same arguments as for the Banach space considerations in [83]. Only the norm properties are required to show convergence identical to the original proof by OJA in [97]. First considering the case of only one principal component, i. e. $K = 1$ and $W = \mathbf{W}_1$. The stationary state is given by $\Delta\mathbf{W} = \mathbf{0}$. Under the assumption of a slowly changing $\mathbf{W}$, i.e. $0 < \eta \ll 1$, the eigenvalue equation

$$\mathrm{E}\left[\mathscr{F}\left[\mathbf{W}\right]\right] = \gamma \cdot \mathbf{W} \tag{4.56}$$

is obtained with $\gamma = \mathrm{E}\left[[\mathbf{V},\mathbf{W}]^*_{S_p} \cdot [\mathbf{V},\mathbf{W}]^*_{S_p}\right]$. As already known, $\mathscr{F}_{\mathbf{V}}\left[\mathbf{W}\right] = [\mathbf{V},\mathbf{W}]^*_{\mathcal{S}_p} \cdot \mathbf{V}$ is a linear operator for each $\mathbf{V} \in \mathcal{V}$, see (3.41) on page 52.
Therefore

$$\begin{aligned}\mathrm{E}\left[\mathscr{F}_{\mathbf{V}}\left[\mathbf{W}\right]\right] &= \mathrm{E}\left[\mathbf{V} \cdot [\mathbf{V},\mathbf{W}]^*_{S_p}\right] && (4.57)\\ &= \mathrm{E}\left[\mathbf{V} \cdot \mathbf{V}^*\right]\left[\mathbf{W}\right] && (4.58)\\ &= \mathbf{C}_{\mathcal{V}}\left[\mathbf{W}\right], && (4.59)\end{aligned}$$

results, which is completely analogously to the considerations of Hebbian PCA learning in separable Banach spaces. The generalization to $K > 1$ is straightforward and follows the argumentation in [115]. This completes the proof. □

4.5 Numerical Simulations and Selected Applications

This subsection starts with example applications and simulations for Non-Euclidean PCA. The different properties of the used inner products, SIPs, and kernels are presented for several data sets. Thereafter an exemplary application for non-Euclidean ICA is demonstrated and shows that the non-Euclidean variant of Hebbian-like ICA is able to extract non-linearly mixed signals. Finally, the different behavior of Hebbian PCA of the matrix approach is compared with the vectorial case for an illustrative example.

The subsections are based on

*M. Lange, M. Biehl and T. Villmann, "*Non-Euclidean Principal Component Analysis by Hebbian Learning*", Neurocomputing 147 (2015), pp. 107-119.[83]*
*M. Biehl, M. Kästner, M. Lange and T. Villmann, "*Non-Euclidean Principal Component Analysis and Oja's Learning Rule - Theoretical Aspects*", in P.A. Estevez and J.C. Principe and P. Zegers, ed., Advances in Self-Organizing Maps: 9th International Workshop WSOM 2012 Santiage de Chile vol. 198, (Berlin: Springer, 2013), pp. 23-34.[14]*

4.5.1 Non-Euclidean PCA for Vectors by Hebbian Learning

Non-Euclidean PCA is demonstrated on four various applications. At first Hebbian PCA learning based on l_1-norm is applied on a two dimensional toy example and a data set of eigenfaces. Thereafter, for a functional data the eigenvectors with the non-Euclidean PCA by Hebbian learning using Sobolev-norms are generated and in the last application the results of kernel Hebbian PCA learning are demonstrated. For all applications the results of the non-Euclidean PCA are compared with the Euclidean ones.

A Two-Dimensional Toy Example

This artificially generated example, where the l_p-norm is used as non-standard measure, demonstrates the non-Euclidean PCA. A circle C_1 of radius $r_{C_1} = 1$ and a ellipse C_2 with minor and major radius $r^1_{C_2} = 1$ and $r^2_{C_2} = 1.2$ in the two-dimensional plane are considered. The first one corresponds exactly the unit circle of the l_2-norm. As mentioned before, other values of the parameter p results in different shapes of the unit circle, see Figure 4.1. Consequently, the principal directions, the red bold arrows in Figure 4.1, of the circle C_1 variegate accordingly to the unit circles.

Based on this observation, additionally the ellipse C_2 is considered, see Figure 4.2. The corresponding principal components, the green bold arrows in this Figure, for the Euclidean case ($p = 2$) coincide with the axes. Unlike, the principal axes for $p = 1$ using the respective SIP $[\mathbf{x}, \mathbf{y}]_{l_1} = \|\mathbf{y}\|_{l_1} \sum_{i=1}^{n} x_i \cdot \mathsf{sign}\,(y_i)$ in Oja's learning rule are different from the axes, provided that for the major radius $r_2 < \sqrt{2}$ remains valid. If $r_2 > \sqrt{2}$, then the principal directions according to the l_1-norm are the same as for the l_2-norm. The presented simulations taking the ellipse borders as inputs and shows exactly this behavior, see Figure 4.2.

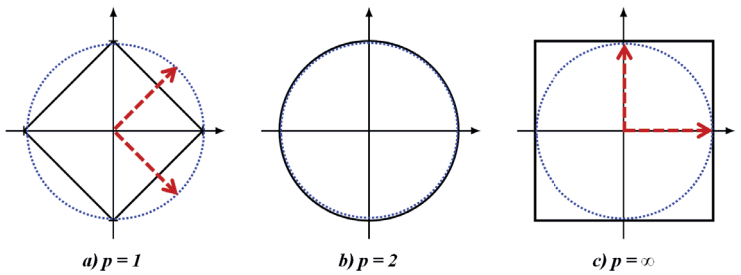

Figure 4.1: Unit balls (black solid lines) and eigenvectors (red bold arrows) for circular data (blue dashed line) for several Minkowski-p-norms $\|\mathbf{x}\|_{l_p}$: a) $p = 1$ - eigenvectors are identical with the diagonals of the rectangular axis system, b) $p = 2$ - no preferred direction, c) $p = \infty$ - the eigenvectors coincide with the axes.

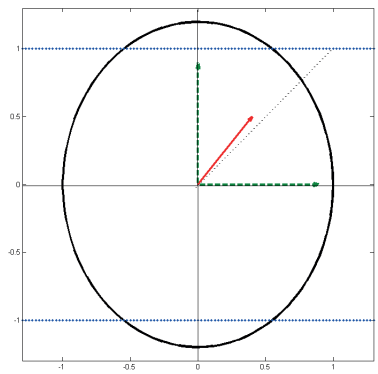

Figure 4.2: Ellipsoid data set with radius $r^1_{C_2} = 1$ and $r^2_{C_2} = 1.2$. The Euclidean eigenvectors coincide with the coordinate axis because the symmetry of the unit circle is broken for an ellipse. The main principal vector according to the l_1-norm (red arrow) differs from the diagonal (dotted) and shifts in the direction given by the major radius r_2. If $r_2 > \sqrt{2}$, then the principal directions according to the l_1-norm coincides with those of the l_2-norm.

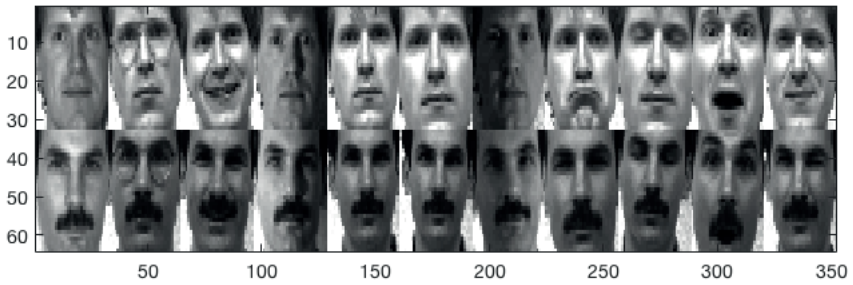

Figure 4.3: Subset of the YALE face recognition data base used in the simulations.

Eigenfaces Using l_p-norms

A more demanding application is the determination of eigenfaces in face recognition [3, 68, 125, 142]. Most frequently, the preferably used method is the standard Euclidean PCA. However, image processing applies commonly the l_1-norm for image comparison [23, 24, 125, 126]. A non-Euclidean PCA with the l_1-norm could be suitable for this application. Thus, the two approaches Sanger's learning rule equipped with the Euclidean inner product and the SIP of the l_1-norm are applied to a subset of the YALE face recognition data base [34] to generate the eigenvectors. This subset contains 32×32 gray level images of two persons with 11 face positions/facial expressions for each, respectively, see figure 4.3, [22]. The resulting eigenfaces (eigenvectors) are illustrated in Figure 4.4. Obviously, the eigenfaces are substantially different. This difference becomes also evident in the reconstructed images of the original faces, see Figure 4.5. Apparently, for this application PCA according to the l_1-norm puts stronger emphasis on contours than the standard Euclidean PCA.

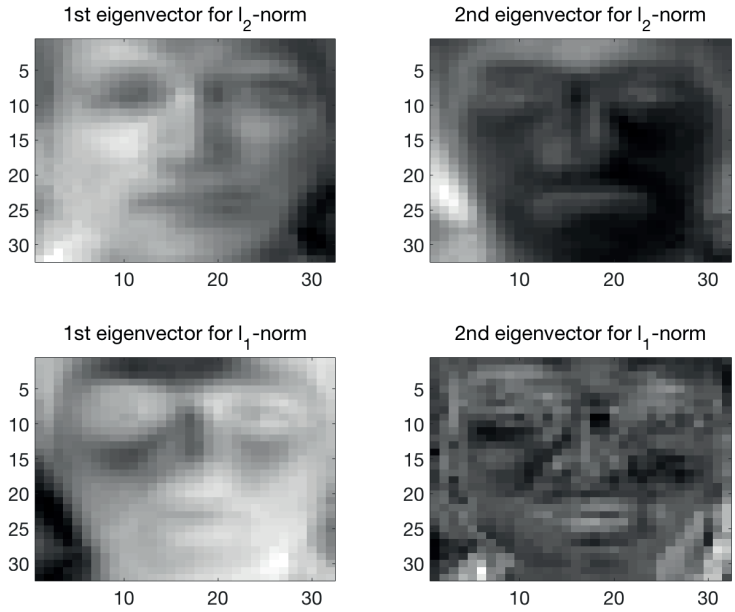

Figure 4.4: First and second eigenfaces obtained for a subset of the YALE face recognition data base using the Euclidean inner product and the SIP $[\mathbf{x}, \mathbf{y}]_{l_1}$ for Sanger's learning rule (see (2.15) on page 14).

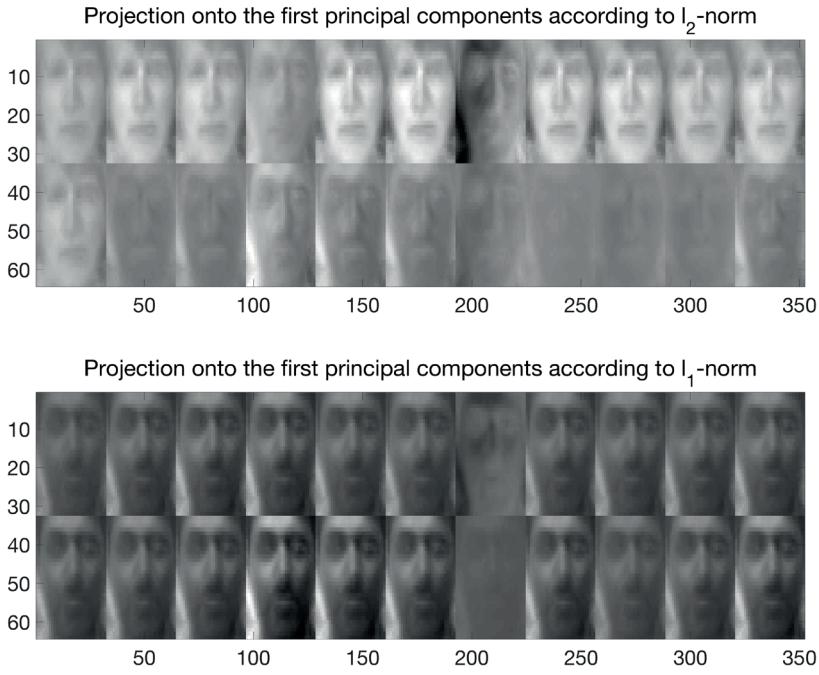

Figure 4.5: Reconstruction of the original face images using only two principal components according to the Euclidean inner product (l_2-norm, top) and the SIP $[\mathbf{x}, \mathbf{y}]_{l_1}$ (l_1-norm bottom). The different behavior is obvious.

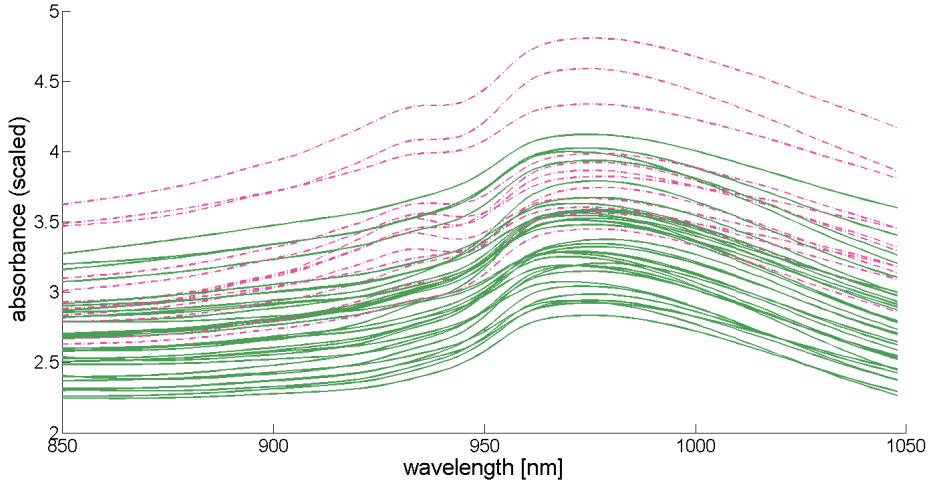

Figure 4.6: Visualization of the TECATOR data set. The data vectors represent smooth spectra of meat probes with high and low fat content (two classes).

Eigenvectors of Functional Data Using Sobolev-Norms

The PCA in the Sobolev space is based on the SIP $[f,g]_{\mathcal{W}_{K,p}} := \frac{1}{\|g\|^{p-2}_{\mathcal{W}_{K,p}}} \sum_{|\alpha|\leq K} \int D^\alpha f \, |D^\alpha g|^{p-1} \, \mathsf{sgn}\,(D^\alpha g)\, dt$ (introduced in chapter 3). Here, functional data $\mathbf{v} \in \mathbb{R}^n$ are vectorial data representing functions, i. e. $v_k = f\,(k)$. It is generally assumed that these functions are smooth and also differentiable.

For demonstration purposes of the Non-Euclidean PCA using Sobolev-norms the TECATOR data set is used [135]. This data set consists of 215 spectra obtained for several meat probes with high and low fat content (two classes), see Figure 4.6. The spectral range of the wavelengths is between 850 nm and 1050 nm.

Three different Non-Euclidean PCA variants by Hebbian learning are applied on TECATOR data set and compared to the results of the Euclidean PCA, i. e., first two eigenvectors according to the largest absolute eigenvalues are generated by Sanger's learning rule based on the l_1-norm, the l_2-norm, the Sobolev-l_1-norm ($\|f\|_{\mathcal{W}_{1,1}}$) and the Sobolev-l_2-norm ($\|f\|_{\mathcal{W}_{1,2}}$), which are visualized in Figure 4.7. The different applied norms result in essential differences. Especially the Sobolev-norm $\|f\|_{\mathcal{W}_{1,1}}$ emphasizes the spectral range around 950 nm, 920 nm and 980 nm. To emphasize the differences between resulted eigenvectors by using the Sobolev-norms and the l_p-

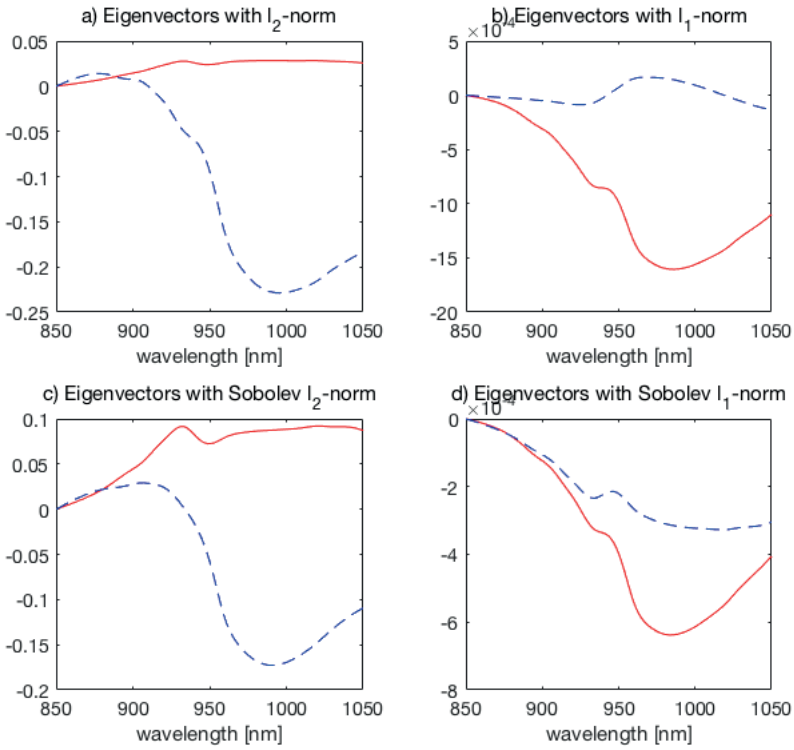

Figure 4.7: For the TECATOR data set the first two eigenvectors for different norms, which are generated by Hebbian PCA learning based on the respective SIPs are visualized: (a) Eigenvectors with the Euclidean norm (b) Eigenvectors with l_1-norm, (c) Eigenvectors with Sobolev-l_2-norm, (d) Eigenvectors with Sobolev-l_1-norm. The Eigenvectors are normalized to unit length according to the respective norm.

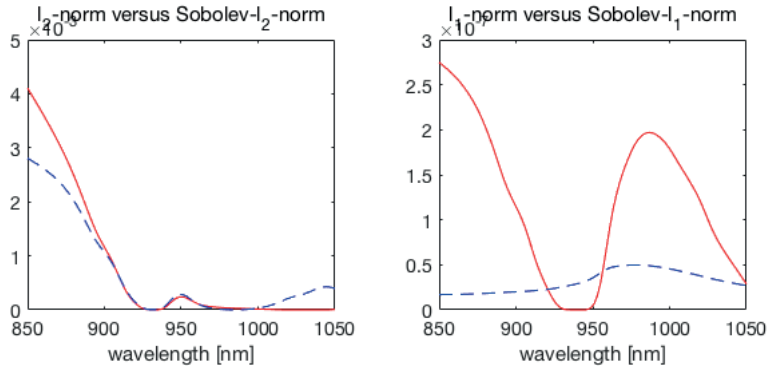

Figure 4.8: Visualization of the quadratic differences for the eigenvectors of PCA between the non-functional l_p-norms and the Sobolev-l_p-norms for the TECATOR data set.

norms the quadratic differences of the obtained eigenvectors are calculated and shown in Figure 4.8. According that the functional Sobolev-norms emphasize curved shapes of data.
The PCA projections of the TECATOR data set according to the applied several norms is visualized in Figure 4.9. According that an improved separation of the two classes can be recognized when using the PCA with the Sobolev-norm $\|f\|_{\mathcal{W}_{1,1}}$ or $\|f\|_{\mathcal{W}_{1,2}}$.
Further, also by relevance learning it turns out that the spectral range around 950 nm are important for classification according to the meat's fat level [66]. The benefit for the classification by using Sobolev-norms in GMLVQ of TECATOR data was also shown in [43]. This benefit is here verified too for the TECATOR data by the PCA projection based on the l_1-norm and the Sobolev-norm $\|f\|_{\mathcal{W}_{1,1}}$, see Figure 4.9.

Eigenvectors in Kernel PCA

The Indian diabetes data set, named PIMA, is a standard data set from UCI, which is frequently applied for the comparison of classifiers [7]. This data set contains 768 data vectors with 8 feature dimensions and is divided into two classes representing the properties "healthy" and "ill". It turned out that learning the classification of these data is relatively difficult. The application of GLVQ based on the Euclidean distance achieves an accuracy of 75.1 % [67]. The accuracy is improved to 78.3 % when using

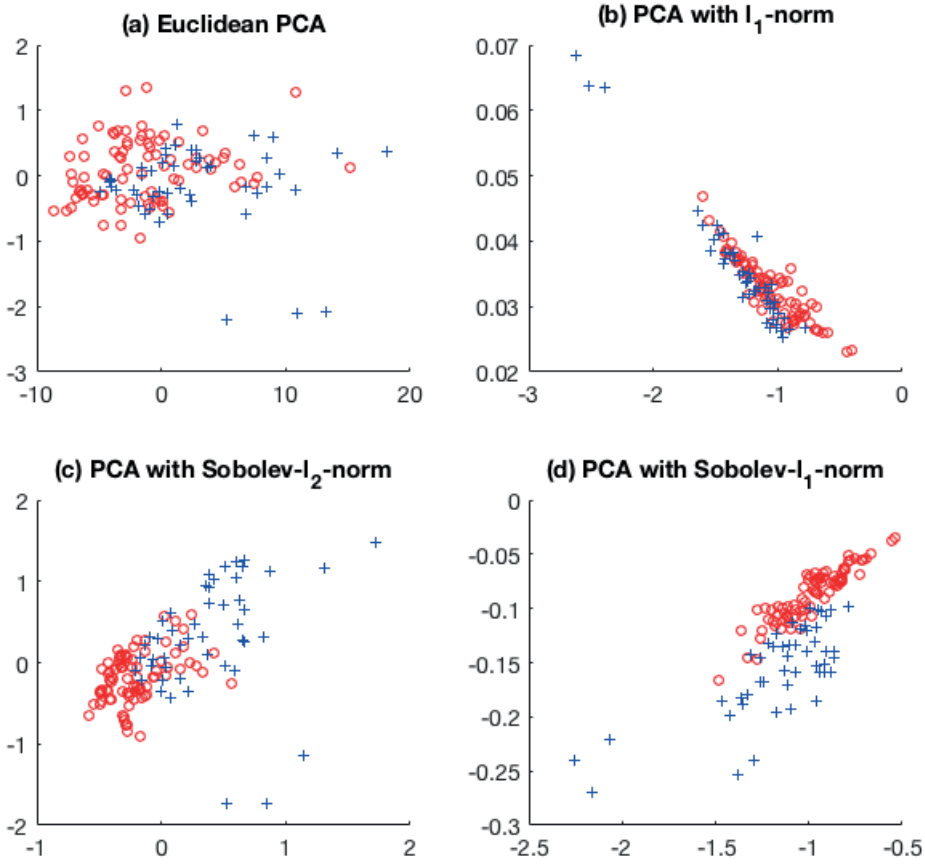

Figure 4.9: Projection of the TECATOR data according to several norms: (a) Euclidean PCA, (b) PCA with l_1-norm, (c) PCA with Sobolev-l_2-norm, (d) PCA with Sobolev-l_1-norm. The classes, i. e. low and high fat content, are displayed as blue crosses and red circles, respectively.

an adaptive exponential kernel distance

$$\kappa_{\Omega}\left(\mathbf{v},\mathbf{w}\right)=\exp\left(-\left(\Omega\left(\mathbf{v}-\mathbf{w}\right)\right)^{2}\right) \tag{4.60}$$

in GLVQ, where Ω is adapted during learning for optimal classification performance. As in the previous example, the Euclidean PCA and the Non-Euclidean PCA for the PIMA data set are compared. Instead of the Euclidean inner product the kernel with the learned matrix Ω, visualized in Figure 4.10, is used to perform a Non-Euclidean PCA, i. e. the first two eigenvectors are obtained by Sanger's learning rule based the kernel (4.60). The eigenvectors and the corresponding PCA projections for the Euclidean and Non-Euclidean variant are visualized in the Figures 4.11 and 4.12, respectively. A slightly improved separability in case of the kernel PCA compared to the Euclidean PCA variant is observed. This results coincides with the improved class separability observed in [67] for kernel distance based classification learning using exactly the same matrix Ω.

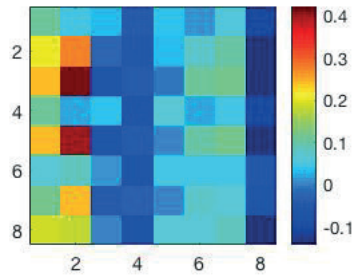

Figure 4.10: Visualization of the kernel matrix Ω in κ_Ω from (4.60) for the PIMA data set.

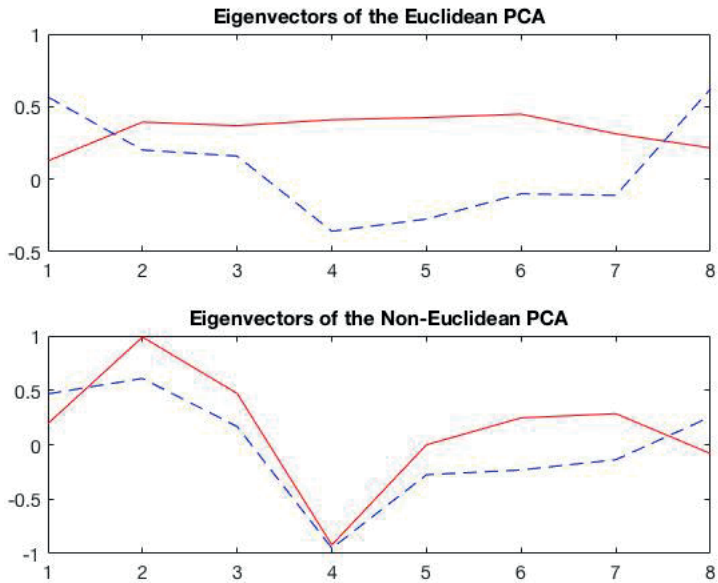

Figure 4.11: Visualization of the eigenvectors of the PIMA data set according to the Euclidean inner product and the kernel from (4.60).

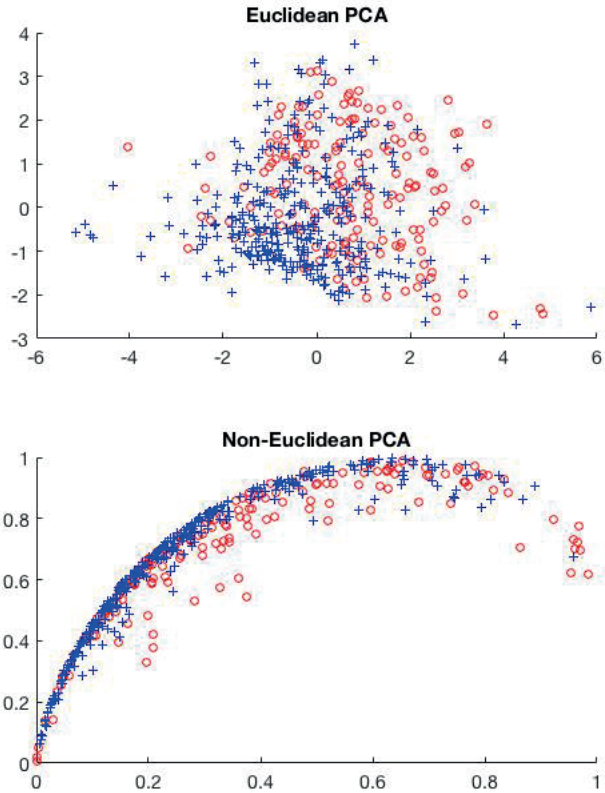

Figure 4.12: The upper picture shows the projection of PIMA data with the Euclidean PCA and the lower picture shows the projection of PIMA data with Oja's Kernel PCA using the kernel κ_Ω from (4.60).

The subsection is based on

M. Lange, M. Biehl, T. Villmann, "Non-Euclidean Independent Component Analysis and Oja's Learning", *in M. Verleysen, ed., Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2013) (Louvain-La-Neuve, Belgium: i6doc.com, 2013), pp. 125-130.[80]*

4.5.2 Non-Euclidean ICA by Hebbian Learning

As an illustrative simulation a non-linear mixture of two source signals is considered, which are generated by linear mixing in the kernel space. The source signals and the resulting mixed signals are visualized in Figure 4.13. The non-linearly mixed signals will be transformed back into the data space by applying the Euclidean and the non-Euclidean variant of Hebbian ICA learning. More precise, the original procedure by Hayvärien&Oja based on the Euclidean inner product, see learning rule (2.31) on page 21, is applied, where for the implicit online estimation of the kurtosis in the second unit an averaging parameter of $\nu = 0.05$ is used. For non-Euclidean demixing of the mixed signals the Euclidean inner product is replaced by the Gaussian Kernel in the learning rule of Hebbian ICA learning, see (4.50) on page 68. The extracted signals generated by the Euclidean and the non-Euclidean variant of Hebbian ICA learning are visualized in Figure 4.13. At first glance the approach with kernels separates out the source signals better than the original Euclidean variant. For verification purposes the squared error and the correlation coefficient of the independent sources are calculated:

	squared error		correlation coefficient	
	signal 1	signal 2	signal 1	signal 2
Euclidean ICA	22.3	55.0	0.86	0.81
Non-Euclidean ICA	23.1	34.8	0.88	0.84

Table 4.1: The squared error (left) and correlation coefficient (right) of the independent sources obtained by the learning rule.

The values given in Table 4.1 show an improved performance of the ICA learning rule based on kernels. This is due to its non-linear character implicitly realized by the kernel trick. However, it is very difficult to stabilize the used kernel ICA algorithm.

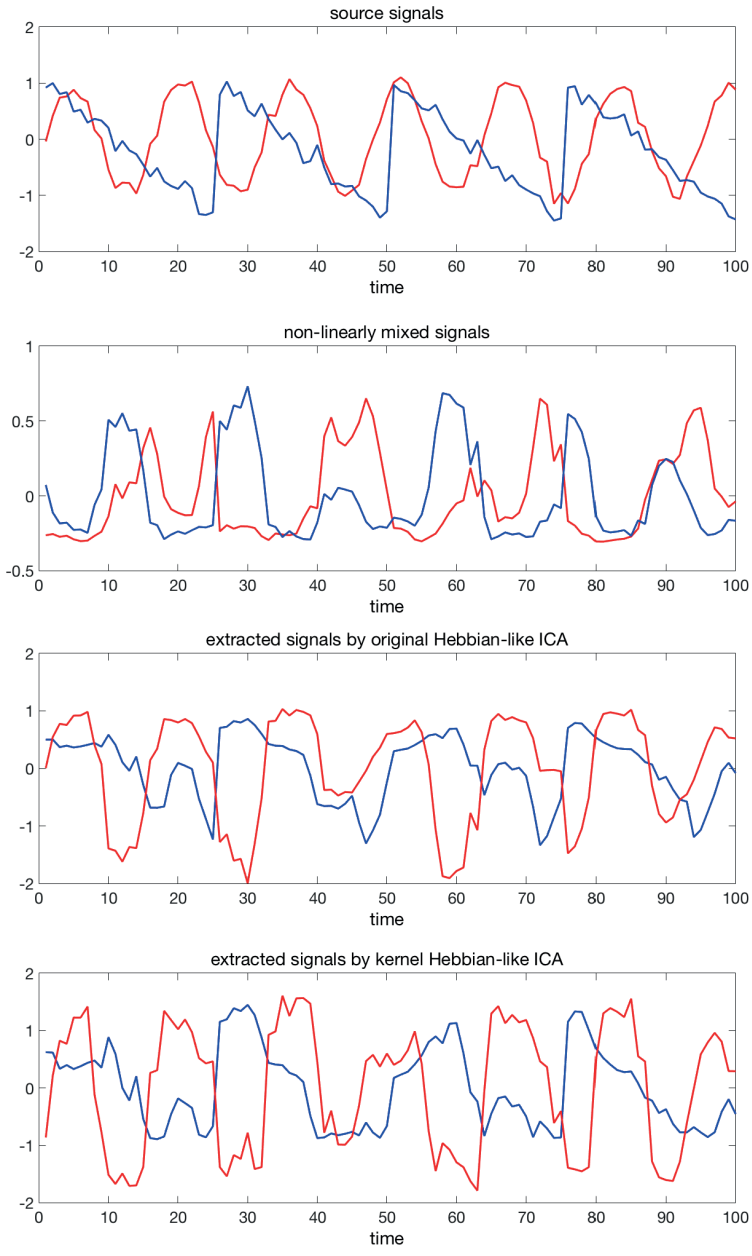

Figure 4.13: Visualization of the original signals (top figure) and the mixed sources (second figure). Estimated source signals obtained by the original Hebbian ICA learning rule using the Euclidean inner product (third figure) and by the kernel variant (bottom figure).

The subsection is based on

M. Lange, D. Nebel and T. Villmann, "Non-Euclidean Principal Component Analysis for Matrices by Hebbian Learning"*, in L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh and J.M. Zurada, ed., Artificial Intelligence and Soft Computing - Proc. the International Conference ICAISC vol. 8467, (Zakopane: Springer, 2014), pp. 77-88.[81]*

A. Villmann, M. Lange-Geisler, T. Villmann, "About Semi-Inner Products for $\mathbf{p-QR}$-Matrix Norms"*, TechReport (2018).[136]*

4.5.3 Non-Euclidean PCA for Matrices by Hebbian Learning

The following experiment, where the Schatten-p-norm is used as non-standard measure, demonstrates the non-Euclidean PCA for matrices by Hebbian learning. The dataset consists of 16×16 gray-scale images of handwritten digits $'0', \ldots, '9'$. The first two principal components according to the largest absolute eigenvalues for several Schatten-p-norms are determined and compared with the respective vectorial counterpart. For the original Hebbian learning variant (vectorial PCA) the matrix data are previously vectorized. The principal components for $p = 1$, $p = 2$ and $p = 5$ of the gray-scale images by using Hebbian PCA Learning for both approaches (for vectors and matrices) are shown in Figures 4.14, 4.15 and 4.16, respectively.

Clear differences between the matrix and the vectorial variant are evident. Thus, the different approaches leads to different principal components, which also caused by the parameter p. The comparisons between the principal components of the vectorial - and matrix approach shows, that the PCA for matrices generates a better performance. Important spatial information may be lost due the vectorization of data, however it is a necessary preprocessing step to apply the original Hebbian PCA (the vectorial variant). The comparison of these PCA results indicates that the data set of the handwritten digits could contain spatial dependencies.

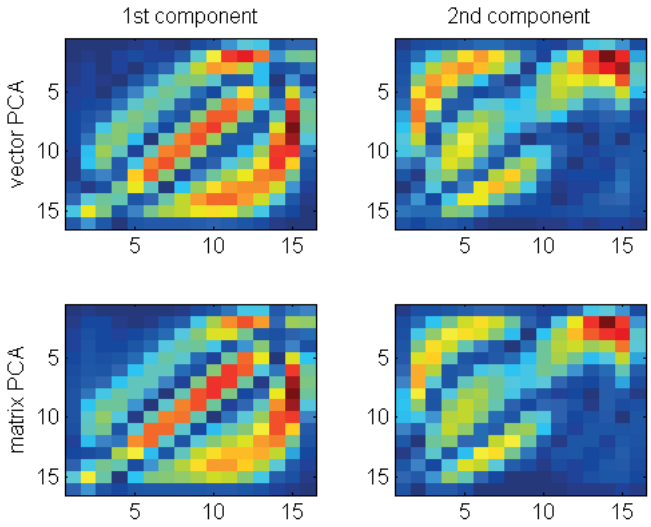

Figure 4.14: Visualization of the first two principal components for the l_2-norm for vectors (top) and matrices (bottom)

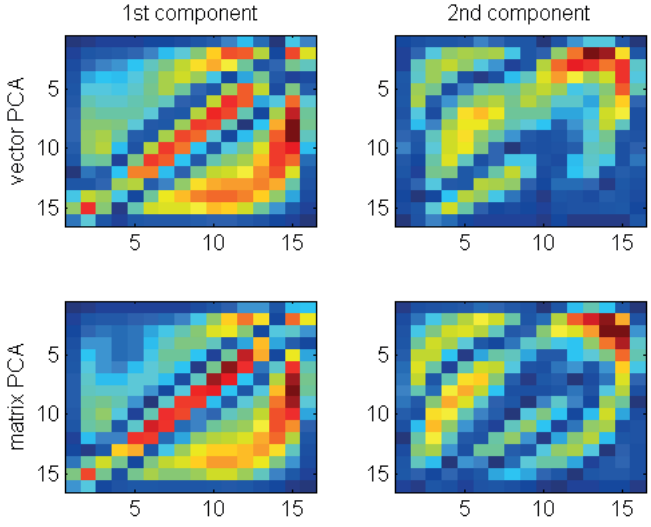

Figure 4.15: Visualization of the first two principal components for the l_1-norm for vectors (top) and matrices (bottom).

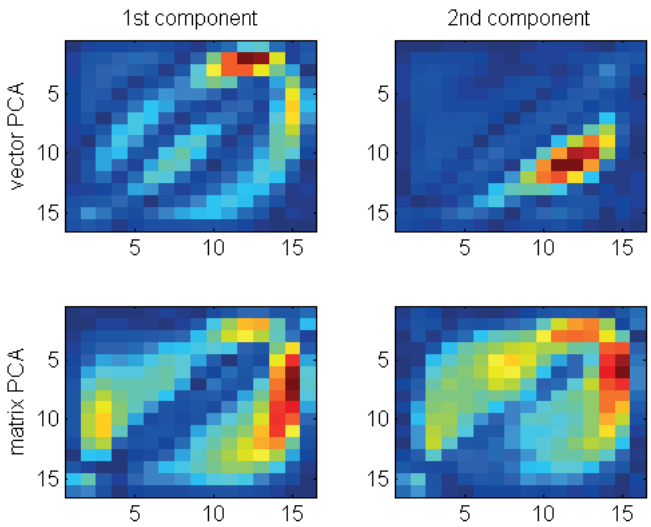

Figure 4.16: Visualization of the first two principal components for the l_5-norm for vectors (top) and matrices (bottom).

Chapter 5

Hebbian Learning Based on General Distance Measures for Variants of Learning Vector Quantization

All LVQ models have in common that they distribute the prototypes in the data space during the learning phase taking into account a dissimilarity measure between data and prototypes. Frequently the squared Euclidean distance is chosen. A more general dissimilarity measure than the squared Euclidean distance can be generated by l_p-norms. In chapter 3 it was emphasized, that l_p-norms show a different behavior depending on the p-values. Thus recently l_p-norms became popular in machine learning approaches [13, 37, 104, 29, 83]. One example is the application of the l_1-norm for noisy data. The larger the p-value the greater the influence of noise for the respective l_p-norm. Hence, for noisy data the l_1-norm is more appropriate than the Euclidean norm. To apply l_p-norms and their induced dissimilarity measures in gradient based learning vector quantization, i. e. GLVQ, their derivatives are required. However, due to the inherent absolute value function in l_p-norms the derivation at the origin ($x = 0$) is not possible except $p = 2$. Therefore, the derivatives require smooth approximations to be applicable in gradient based machine learning approaches. In this chapter the formal derivatives of dissimilarity measures are provided using l_p-norms

as well as their smooth numerical approximation.

A further main point of this chapter is the extension of LVQ approaches to classify matrix data such as functional data depending on time and frequency. Most machine learning classifiers were process usually vectorial data. Classification of matrix data is often executed by vectorization of the data in advance. For images a huge number of sophisticated methods, such as feature extraction, are available [26], as mentioned in the chapter before. Most frequently, the generated features are collected in a vector, which is processed instead of the images. If the two dimensions of the functional data matrix cannot be factorized, any applied vectorization method leads to a loss of information. Thus, a direct processing of matrix data is needed. Therefore, an extension of LVQ, called learning matrix quantization (LMQ), is proposed to classify matrix data avoiding vectorization or feature extraction. At first glance the extension is straight forward due to the fact that matrix data also generates a vector space. The vector norm (l_p-norm) is simply replaced by a appropriate norm for matrix processing. One of the most applied matrix norms and known to be suitable for dissimilarity determination of many matrix data problems is the Schatten-p-norm [117]. Schatten-p-norms can be seen as generalizations of l_p-norms. The mathematical foundations of Schatten-p-norms were introduced in chapter 3.6. The application of distances generated by Schatten-p-norms in LMQ offers several variants of relevance learning. Certainly the benefit of LMQ is the higher flexibility of relevance learning in comparison to LVQ but it is usually more complicate and in general the interpretation is difficult. In this chapter several variants of LMQ are discussed together with exemplary applications to show their different behaviors are presented.

The chapter is structured as follows: It starts with learning based on vector norms presenting the derivatives of the dissimilarity measure by using l_p-norms and their smooth numerical approximations. Afterwards, the learning based on matrix norms, (LMQ) is considered. Finally, numerical simulations and selected applications of LVQ based on l_p-norms and the LMQ algorithm applying the Schatten-p-norm for $p = 2$ together with several possibilities of relevance learning are presented.

The subsections are based on

M. Lange, T. Villmann, "Derivatives of l_p-norms and their Approximations", *Machine Learning Reports* 7, *MLR-04-2013 (2013), pp. 43-59. [79]*
M. Lange, D. Zühlke, O. Holz, T. Villmann, "Applications of l_p-norms and their Smooth Approximations for Gradient Based Learning Vector Quantization", *in M. Verleysen, ed., Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2014),pp. 271-276. [82]*
*M. Kaden, M. Lange, D. Nebel, M. Riedel and T. Geweniger and T. Villmann, "*Aspects in Classification Learning - Review of Recent Developments in Learning Vector Quantization*", Foundations of Computing and Decision Sciences 39 (2014), pp. 79-105.[64]*

5.1 Learning Based on Vector Norms

First the formal derivatives of the dissimilarity measures using l_p-norms are provided. Then, required smooth numerical approximations for the maximum- and absolute value function as well as their derivatives are introduced. Additionally, a brief remark about use of Laplacian and l_p-kernels in gradient based methods is given.

5.1.1 l_p-norms and their Derivatives

Let $\mathbf{v} \in \mathbb{R}^n$ be data vectors and $\mathbf{w} \in \mathbb{R}^n$ be prototypes. Depending on the selected p value the dissimilarity measure generated by the l_p-norm becomes $d_p(\mathbf{v}, \mathbf{w}) = \sum_{i=1}^{n} |v_i - w_i|^p$ for $0 < p < \infty$ and in case of $p = \infty$ the dissimilarity measure reads as $d_\infty(\mathbf{v}, \mathbf{w}) = \max(|\mathbf{v} - \mathbf{w}|)$. The formal derivatives of both cases are considered separately.

Formal Derivatives of l_p-norms for $0 < p < \infty$

For $0 < p < \infty$ the optimization of a prototype using $d_p(\mathbf{v}, \mathbf{w}) = \sum_{i=1}^{n} |v_i - w_i|^p$ requires the formal derivative

$$\frac{\partial d_p(\mathbf{v}, \mathbf{w})}{\partial w_k} = -p \cdot |v_k - w_k|^{p-1} \cdot \frac{\partial |v_k - w_k|}{\partial w_k}. \tag{5.1}$$

The vectorial form of the gradient (5.1) is

$$\frac{\partial d_p(\mathbf{v}, \mathbf{w})}{\partial \mathbf{w}} = -p \cdot |\mathbf{v} - \mathbf{w}|^{\star(p-1)} \circ \frac{\partial |\mathbf{v} - \mathbf{w}|}{\partial \mathbf{w}}, \tag{5.2}$$

where $\mathbf{x} \circ \mathbf{y} = (x_1 \cdot y_1, \ldots, x_n \cdot y_n)^\top$ denotes the Hadamard product and $\mathbf{x}^{\star k}$ the component wise power according to $\mathbf{x}^{\star k} = \left(x_1^k, \ldots, x_n^k\right)^\top$. For the weighted dissimilarity measure $d_{p,\boldsymbol{\lambda}}(\mathbf{v}, \mathbf{w}) = \sum_{i=1}^n \lambda_i |v_i - w_i|^p$ the gradient can be taken as

$$\frac{\partial d_{p,\boldsymbol{\lambda}}(\mathbf{v}, \mathbf{w})}{\partial \mathbf{w}} = -p \cdot \boldsymbol{\lambda} \circ |\mathbf{v} - \mathbf{w}|^{\star(p-1)} \circ \frac{\partial |\mathbf{v} - \mathbf{w}|}{\partial \mathbf{w}} \tag{5.3}$$

whereas

$$\frac{\partial d_{p,\boldsymbol{\lambda}}(\mathbf{v}, \mathbf{w})}{\partial \boldsymbol{\lambda}} = |\mathbf{v} - \mathbf{w}|^{\star p} \tag{5.4}$$

is the derivative for $\boldsymbol{\lambda}$. The gradients in GMLVQ using the matrix variant $d_{p,\boldsymbol{\Omega}}(\mathbf{v}, \mathbf{w}) = \sum_{i=1}^m \left| \sum_{j=1}^n \Omega_{i,j} (v_j - w_j) \right|^p$ result in

$$\frac{\partial d_{p,\boldsymbol{\Omega}}(\mathbf{v}, \mathbf{w})}{\partial w_k} = -p \sum_{i=1}^m \left| \sum_{j=1}^n \Omega_{i,j} (v_j - w_j) \right|^{\star(p-1)} \cdot \frac{\partial \left| \sum_{j=1}^n \Omega_{i,j} (v_j - w_j) \right|}{\partial \sum_{j=1}^n \Omega_{i,j} (v_j - w_j)} \cdot \Omega_{i,k} \tag{5.5}$$

with $\frac{\partial \sum_{j=1}^n \Omega_{i,j}(v_j - w_j)}{\partial w_k} = -\Omega_{i,k}$ and

$$\frac{\partial d_{p,\boldsymbol{\Omega}}(\mathbf{v}, \mathbf{w})}{\partial \Omega_{kl}} = p \left(\left| \sum_{j=1}^n \Omega_{k,j} (v_j - w_j) \right|^{\star(p-1)} \circ \frac{\partial \left| \sum_{j=1}^n \Omega_{k,j} (v_j - w_j) \right|}{\partial \sum_{j=1}^n \Omega_{k,j} (v_j - w_j)} \right) \cdot (v_l - w_l), \tag{5.6}$$

with $\frac{\partial \sum_{j=1}^n \Omega_{k,j}(v_j - w_j)}{\partial \Omega_{k,l}} = (v_l - w_l)$. The vector notation of both derivatives (5.5) and (5.6) read as

$$\frac{\partial d_{p,\boldsymbol{\Omega}}(\mathbf{v}, \mathbf{w})}{\partial \mathbf{w}} = -p \cdot \boldsymbol{\Omega}^\top \left(|\boldsymbol{\Omega}(\mathbf{v} - \mathbf{w})|^{\star(p-1)} \circ \frac{\partial |\boldsymbol{\Omega}(\mathbf{v} - \mathbf{w})|}{\partial \mathbf{w}} \right) \tag{5.7}$$

and

$$\frac{\partial d_{p,\boldsymbol{\Omega}}(\mathbf{v}, \mathbf{w})}{\partial \boldsymbol{\Omega}} = p \left(|\boldsymbol{\Omega}(\mathbf{v} - \mathbf{w})|^{\star(p-1)} \circ \frac{\partial |\boldsymbol{\Omega}(\mathbf{v} - \mathbf{w})|}{\partial \mathbf{w}} \right) \cdot (\mathbf{v} - \mathbf{w})^\top, \tag{5.8}$$

where it is assumed that vectors are column vectors. All following derivatives are stated in vector notation for better readability. The associated element wise derivatives are given in the Appendix B.8.

Formal Derivatives of l_p-norms for $p = \infty$

As previously introduced, the l_p-norm for $p = \infty$ becomes the maximum function $\max(\mathbf{x}) = \max_i(x_i)$. Therefore, the derivatives in GLVQ, GRLVQ and GMLVQ include the maximum- as well as the absolute value function. Then, the optimization of a prototype using $d_\infty(\mathbf{v}, \mathbf{w}) = \max(|\mathbf{v} - \mathbf{w}|)$ yields the formal derivative

$$\frac{\partial d_\infty(\mathbf{v}, \mathbf{w})}{\partial \mathbf{w}} = \frac{\partial \max(|\mathbf{v} - \mathbf{w}|)}{\partial(\mathbf{v} - \mathbf{w})} \cdot \frac{\partial |\mathbf{v} - \mathbf{w}|}{\partial(\mathbf{v} - \mathbf{w})} \cdot \frac{\partial(\mathbf{v} - \mathbf{w})}{\partial \mathbf{w}} \tag{5.9}$$

$$= -\frac{\partial \max(|\mathbf{v} - \mathbf{w}|)}{\partial(\mathbf{v} - \mathbf{w})} \cdot \frac{\partial |\mathbf{v} - \mathbf{w}|}{\partial(\mathbf{v} - \mathbf{w})}. \tag{5.10}$$

Analogously, the prototype update of the weighted variant $d_{\infty,\boldsymbol{\lambda}}(\mathbf{v}, \mathbf{w}) = \max(\boldsymbol{\lambda} \circ |\mathbf{v} - \mathbf{w}|)$ requires the formal derivative

$$\frac{\partial d_{\infty,\boldsymbol{\lambda}}(\mathbf{v}, \mathbf{w})}{\partial \mathbf{w}} = -\frac{\partial \max(\boldsymbol{\lambda} \circ |\mathbf{v} - \mathbf{w}|)}{\partial(\boldsymbol{\lambda} \circ |\mathbf{v} - \mathbf{w}|)} \cdot \frac{\partial(\boldsymbol{\lambda} \circ |\mathbf{v} - \mathbf{w}|)}{\partial(\mathbf{v} - \mathbf{w})} \tag{5.11}$$

whereas the prototype update of the matrix variant $d_{\infty,\boldsymbol{\Omega}}(\mathbf{v}, \mathbf{w}) = \max(|\boldsymbol{\Omega}(\mathbf{v} - \mathbf{w})|)$ involves

$$\frac{\partial d_{\infty,\boldsymbol{\Omega}}(\mathbf{v}, \mathbf{w})}{\partial \mathbf{w}} = -\boldsymbol{\Omega}^\top \frac{\partial \max(|\boldsymbol{\Omega}(\mathbf{v} - \mathbf{w})|)}{\partial \boldsymbol{\Omega}(\mathbf{v} - \mathbf{w})} \cdot \frac{\partial(|\boldsymbol{\Omega}(\mathbf{v} - \mathbf{w})|)}{\partial \boldsymbol{\Omega}(\mathbf{v} - \mathbf{w})}. \tag{5.12}$$

The optimization of the relevance vector $\boldsymbol{\lambda}$ and matrix $\boldsymbol{\Omega}$ takes place according the derivatives

$$\frac{\partial d_{\infty,\boldsymbol{\lambda}}(\mathbf{v}, \mathbf{w})}{\partial \boldsymbol{\lambda}} = \frac{\partial \max(\boldsymbol{\lambda} \circ |\mathbf{v} - \mathbf{w}|)}{\partial(\boldsymbol{\lambda} \circ |\mathbf{v} - \mathbf{w}|)} \cdot |\mathbf{v} - \mathbf{w}| \tag{5.13}$$

and

$$\frac{\partial d_{\infty,\boldsymbol{\Omega}}(\mathbf{v}, \mathbf{w})}{\partial \boldsymbol{\Omega}} = \frac{\partial \max(|\boldsymbol{\Omega}(\mathbf{v} - \mathbf{w})|)}{\partial \boldsymbol{\Omega}(\mathbf{v} - \mathbf{w})} \cdot \frac{\partial(|\boldsymbol{\Omega}(\mathbf{v} - \mathbf{w})|)}{\partial(\mathbf{v} - \mathbf{w})} \cdot (\mathbf{v} - \mathbf{w})^\top, \tag{5.14}$$

respectively. The just considered formal derivatives of the l_p-norm require the derivatives of the maximum- and the absolute value function, which are not differentiable functions. Therefore, smooth approximations of them are introduced in the following.

5.1.2 Smooth Numerical Approximations for the Maximum Function and Absolute Value Function and their Derivatives

As already mentioned, smooth approximations of the absolute value function and the maximum function are required to use dissimilarity measures based on l_p-norms in gradient based methods. In the following two different differentiable approximation functions and their derivatives are explained and their differences are discussed.

Smooth Approximations of the Maximum Function

One appropriate variant to approximate the maximum function $\max(\mathbf{x}) = \max_i(x_i)$ is the α-*softmax* function

$$\mathcal{S}_\alpha(\mathbf{x}) = \frac{\sum_{i=1}^n x_i e^{\alpha x_i}}{\sum_{i=1}^n e^{\alpha x_i}} \tag{5.15}$$

with $\alpha > 0$. $\mathcal{S}_\alpha$ is frequently applied in optimization and neural computation [20, 46] as well as in deep learning [57, 12, 11]. If $\alpha < 0$ is chosen in (5.15), then $\mathcal{S}_\alpha$ becomes a smooth approximation of the minimum function $\min(\mathbf{x}) = \min_i(x_i)$.
Another smooth approximation of the maximum function, proposed by J.D. Cook in [29], is defined as

$$\mathcal{Q}_\alpha(\mathbf{x}) = \frac{1}{\alpha}\log\left(\sum_{i=1}^n e^{\alpha x_i}\right), \tag{5.16}$$

which is related to the α-softmax function. The approximation $\mathcal{Q}_\alpha(\mathbf{x})$ is termed as α-*quasimax* function. According to [75] the α-*quasimax* function can be understood as a kind of generalized functional mean or quasi arithmetic mean. Further, note that

$$\mathcal{Q}_\alpha(\mathbf{x}) \leq \max(\mathbf{x}) + \frac{\log(n)}{\alpha} \tag{5.17}$$

is valid. Both approximation functions include an α-parameter, which controls the precision of the approximation of the maximum function. A small α yields a poor approximation whereas large α-values is yield a good approximation of the maximum function, see Figure 5.1.However, for larger α-values $\mathcal{S}_\alpha$ shows an overshooting of the principle diagonals in contrast to $\mathcal{Q}_\alpha$. These differences are also obvious by considering the deviation of approximation function and the maximum function, i.e. $\mathcal{S}_\alpha(\mathbf{x}) - \max(\mathbf{x})$ and $\mathcal{Q}_\alpha(\mathbf{x}) - \max(\mathbf{x})$, pictured in Figure 5.2. The region sensitive to the overshooting behavior around the principal diagonal of $\mathcal{S}_\alpha$ degreases with increasing α-values, but is still larger for $\mathcal{S}_\alpha$ then for $\mathcal{Q}_\alpha$.

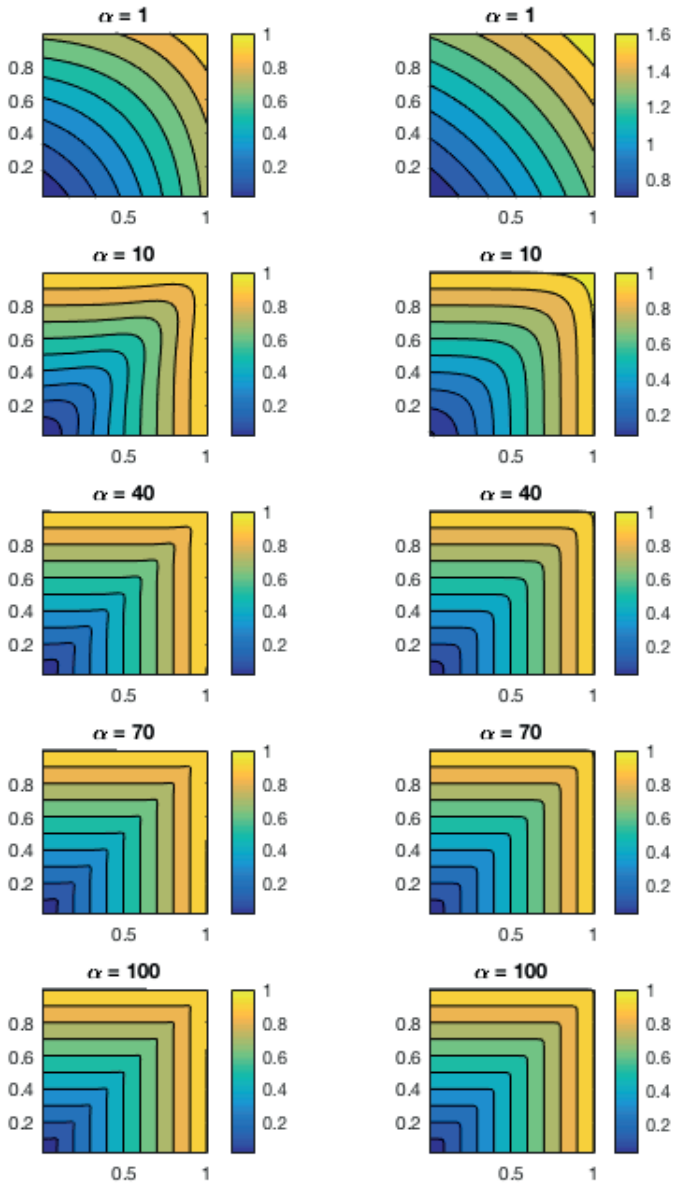

Figure 5.1: Visualization of smooth approximations for the maximum function. Left column shows the softmax function $\mathcal{S}_\alpha$ for different α values and the right column the quasimax function $\mathcal{Q}_\alpha$.

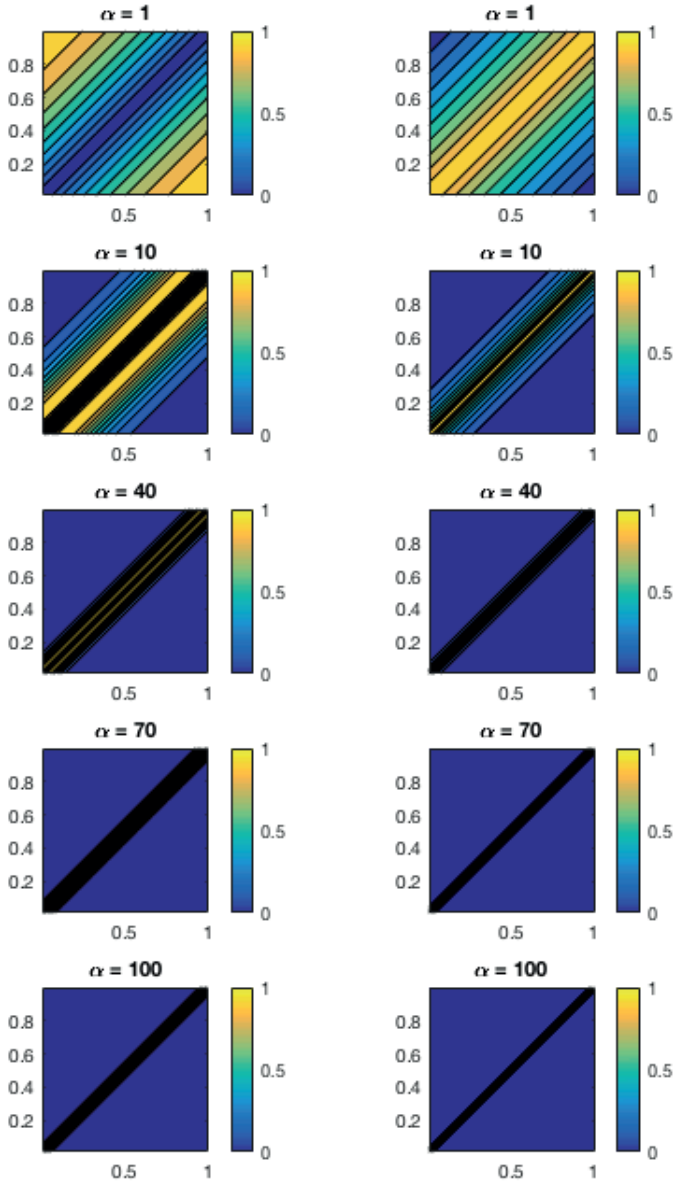

Figure 5.2: Left column shows the difference between the softmax function $\mathcal{S}_\alpha(x)$ and $\max(x)$ for different $\alpha-$values and the right column the difference between the quasimax function $\mathcal{Q}_\alpha$ and $\max(x)$.

Derivatives of Smooth Numerical Approximations of the Maximum Function

To apply the above introduced smooth approximations of the maximum in gradient based numerical methods, neural networks, or other methods in machine learning the corresponding derivatives are required. The gradient of the α-*softmax* function $\mathcal{S}_\alpha$ (5.15) can be expressed in terms of $\mathcal{S}_\alpha$ itself, i. e.

$$\frac{\partial \mathcal{S}_\alpha(\mathbf{x})}{\partial x_k} = \frac{e^{\alpha x_k}}{\sum_{i=1}^{n} e^{\alpha x_i}} \left[1 + \alpha \left(x_k - S_\alpha(\mathbf{x})\right)\right] \tag{5.18}$$

whereas the derivative of the α-*quasimax* function $\mathcal{Q}_\alpha$ (5.16) becomes

$$\frac{\partial \mathcal{Q}_\alpha(\mathbf{x})}{\partial x_k} = \frac{e^{\alpha x_k}}{\sum_{i=1}^{n} e^{\alpha x_i}}. \tag{5.19}$$

Obviously, the derivatives $\frac{\partial \mathcal{S}_\alpha(\mathbf{x})}{\partial x_k}$ and $\frac{\partial \mathcal{Q}_\alpha(\mathbf{x})}{\partial x_k}$ look similar apart from a slight variation and, hence, $\frac{\partial \mathcal{S}_\alpha(\mathbf{x})}{\partial x_k}$ can be written in terms of $\frac{\partial \mathcal{Q}_\alpha(\mathbf{x})}{\partial x_k}$

$$\frac{\partial \mathcal{S}_\alpha(\mathbf{x})}{\partial x_k} = \frac{\partial \mathcal{Q}_\alpha(\mathbf{x})}{\partial x_k} \nabla_{\mathcal{SQ}}, \tag{5.20}$$

where

$$\nabla_{\mathcal{SQ}} = \left[1 + \alpha \left(x_k - S_\alpha(\mathbf{x})\right)\right]. \tag{5.21}$$

The effect of $\nabla_{\mathcal{SQ}}$ in the derivative is emphasized in Figure 5.3. The comparison of the left and right column shows that $\nabla_{\mathcal{SQ}}$ causes a nonlinear behavior in the derivative of $\mathcal{S}_\alpha(\mathbf{x})$. Further, the above mentioned slight differences regarding the range around the principal diagonals between both approximation functions is also evident in the derivatives. In summary, it can be concluded that the α-*quasimax* function better approximating the maximum function and its critical range is smaller compared to the one of the α-*softmax* function.

Consistent Smooth Approximations of the Absolute Value Function

Smooth approximations of the absolute value function are based on the maximum function. One variant, suggested in [118], approximates the absolute value by

$$|x|_\alpha = (x)_\alpha^+ + (-x)_\alpha^+, \tag{5.22}$$

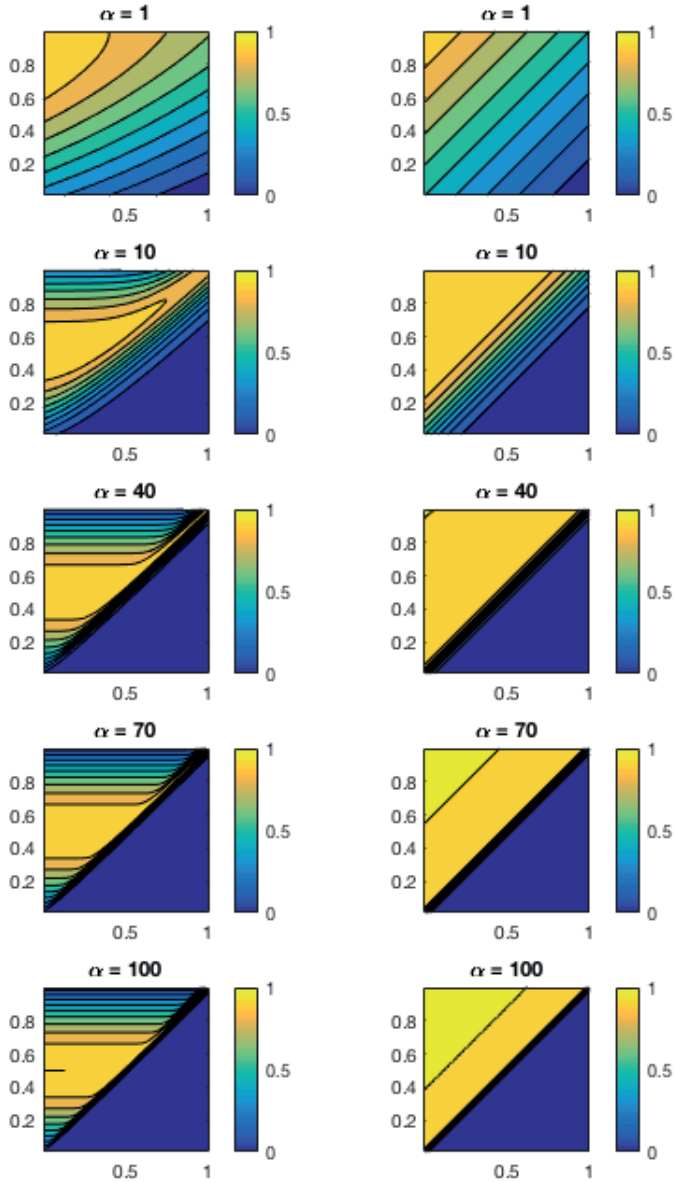

Figure 5.3: Left column shows the derivatives of the softmax function $\mathcal{S}_\alpha(x)$ for different α values and in the right column the derivative of the quasimax function $\mathcal{Q}_\alpha$ are presented.

where

$$(x)_\alpha^+ = \max(\mathbf{x}_0) \tag{5.23}$$

with $\mathbf{x}_0 = (x, 0)^\top$. In [25] the maximum function is replaced by a convex α-approximation

$$(x)_\alpha^+ = x + \frac{1}{\alpha}\log\left(1 + e^{-\alpha x}\right) \tag{5.24}$$

defined for values $x > 0$, which is related to the previously introduced α-*quasimax* function $\mathcal{Q}_\alpha(\mathbf{x})$ (5.16). In detail, taking into account that $e^{-\alpha\cdot 0} = 1$, equation (5.24) can be written in terms of the α-*quasimax* function $\mathcal{Q}_\alpha$ (5.16)

$$(x)_\alpha^+ = x + \mathcal{Q}_\alpha(-\mathbf{x}_0). \tag{5.25}$$

thus, the absolute value function (5.22) can be approximated by

$$|x|_\alpha^{\mathcal{Q}} = \frac{1}{\alpha}\log\left(2 + e^{-\alpha x} + e^{\alpha x}\right), \tag{5.26}$$

which is based on $\mathcal{Q}_\alpha$ and, hence, referred to as α-*quasi-absolute* function $|x|_\alpha^{\mathcal{Q}}$. Because $|x|_\alpha^{\mathcal{Q}}$ is consistent with $\mathcal{Q}_\alpha(\mathbf{x})$ the upper bound

$$\left||x| - |x|_\alpha^{\mathcal{Q}}\right| \leq 2\frac{\log(2)}{\alpha} \tag{5.27}$$

is valid, which is shown in [108].

In equation (5.25) another smooth approximation for the maximum function can be used instead of $\mathcal{Q}_\alpha$. If $\mathcal{Q}_\alpha$ is replaced by the α-*softmax* function $\mathcal{S}_\alpha$ (5.15) in

(5.25), then the absolute value function can be approximated by the α-*soft-absolute* function

$$|x|_\alpha^S = \frac{x\cdot(e^{\alpha x} + e^{-\alpha x})}{2 + e^{\alpha x} + e^{-\alpha x}} \tag{5.28}$$

with

$$(x)_\alpha^+ = x + \mathcal{S}_\alpha(\mathbf{x}_0). \tag{5.29}$$

Both approximations $|x|_\alpha^{\mathcal{Q}}$ and $|x|_\alpha^S$ are depicted in Figure 5.4 for several α-values. The α-values influence again the precision of the approximating functions. Small α-values lead to a poor approximation whereas large α-values yield a better approximation of the absolute value function, which is analogous to the smooth approximation functions of the maximum. Also, the main differences between $\mathcal{S}_\alpha$ and $\mathcal{Q}_\alpha$, which were previously discussed and visualized, are obviously transferred to $|x|_\alpha^{\mathcal{S}}$ and $|x|_\alpha^{\mathcal{Q}}$.

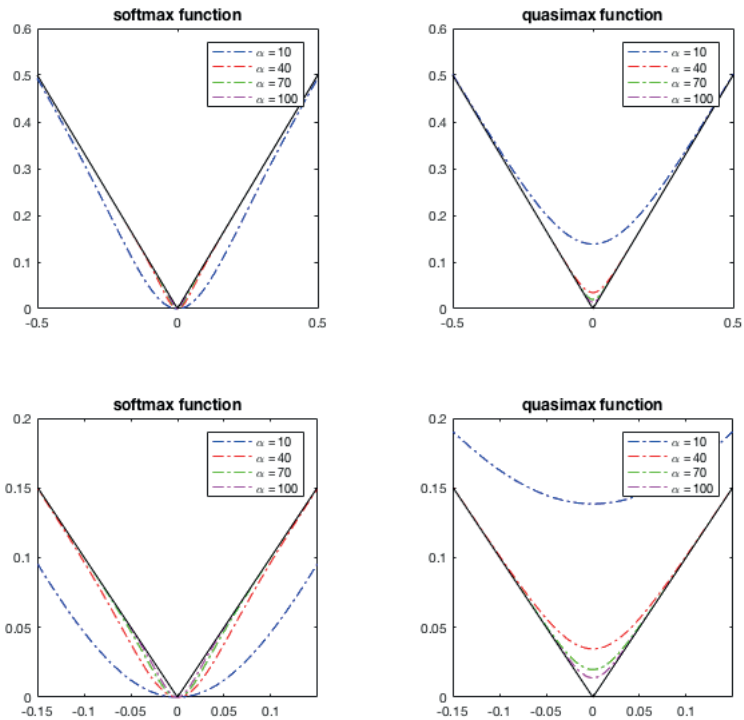

Figure 5.4: The approximation functions of $|x|_\alpha^S$ (left column) and $|x|_\alpha^{\mathcal{Q}}$ (right column) for different α values are depicted. The colored curves are the smooth approximations and the black ones are the absolute value functions. The second row magnifies the problem areas of the approximation functions for better visualization of the different behavior of the approximation functions.

Derivatives of Smooth Numerical Approximations of the Absolute Value Function

To apply the smooth approximations $|x|_\alpha^{\mathcal{S}}$ and $|x|_\alpha^{\mathcal{Q}}$ in gradient based methods the derivatives are also required. The derivatives of the α-*quasi-absolute* function $|x|_\alpha^{\mathcal{Q}}$ and α-*soft-absolute* function $|x|_\alpha^{S}$ result in

$$\frac{\partial |x|_\alpha^{\mathcal{Q}}}{\partial x} = \frac{e^{\alpha x} - e^{-\alpha x}}{2 + e^{\alpha x} + e^{-\alpha x}} \tag{5.30}$$

$$= \tanh\left(\frac{\alpha}{2} x\right) \tag{5.31}$$

and

$$\frac{\partial |x|_\alpha^{\mathcal{S}}}{\partial x} = \frac{x\left(e^{\alpha x} - e^{-\alpha x}\right)}{2 + e^{\alpha x} + e^{-\alpha x}} + x \tag{5.32}$$

$$= \tanh\left(\frac{\alpha}{2} x\right) + \frac{\alpha x}{2\left(\cosh\left(\frac{\alpha}{2} x\right)\right)^2}, \tag{5.33}$$

respectively. Although $|x|_\alpha^{\mathcal{Q}}$ and $|x|_\alpha^{S}$ look different, their derivatives differ by only an additive term

$$\frac{\partial |x|_\alpha^{\mathcal{S}}}{\partial x} = \frac{\partial |x|_\alpha^{\mathcal{Q}}}{\partial x} + \triangle_{\mathcal{SQ}}(\alpha x) \tag{5.34}$$

with the deviation term

$$\triangle_{\mathcal{SQ}}(\alpha x) = \frac{\alpha x}{2\left(\cosh\left(\frac{\alpha}{2} x\right)\right)^2}. \tag{5.35}$$

For the application of a dissimilarity induced by l_p-norms in gradient based LVQ care should be taken to ensure that the underlying approximation functions are consistent. That means, if the derivative of $d_p(\mathbf{v}, \mathbf{w})$ includes the derivatives of the maximum function and the absolute value function then the used approximation functions should be consistent, i. e.

$$\frac{\partial d_\infty(\mathbf{v}, \mathbf{w})}{\partial \mathbf{w}} = -\frac{\partial \mathcal{S}_\alpha(|\mathbf{v} - \mathbf{w}|)}{\partial(\mathbf{v} - \mathbf{w})} \cdot \frac{\partial |\mathbf{v} - \mathbf{w}|_\alpha^{\mathcal{S}}}{\partial(\mathbf{v} - \mathbf{w})} \tag{5.36}$$

or

$$\frac{\partial d_\infty(\mathbf{v}, \mathbf{w})}{\partial \mathbf{w}} = -\frac{\partial \mathcal{Q}_\alpha(|\mathbf{v} - \mathbf{w}|)}{\partial(\mathbf{v} - \mathbf{w})} \cdot \frac{\partial |\mathbf{v} - \mathbf{w}|_\alpha^{\mathcal{Q}}}{\partial(\mathbf{v} - \mathbf{w})} \tag{5.37}$$

are consistent updates of (5.9) for $p = \infty$.

Laplacian and l_p-Kernels

The l_p-norms are also involved in several exponential kernels. The most familiar exponential kernels are the Laplacian Kernel

$$L(\mathbf{v}, \mathbf{w}) = e^{-d_1(\mathbf{v},\mathbf{w})}, \tag{5.38}$$

the Gaussian Kernel

$$G(\mathbf{v}, \mathbf{w}) = e^{-\frac{d_2(\mathbf{v},\mathbf{w})}{2\sigma^2}} \tag{5.39}$$

and the exponential Kernel

$$\widetilde{G}(\mathbf{v}, \mathbf{w}) = e^{-\frac{d_2^*(\mathbf{v},\mathbf{w})}{2\sigma^2}} \tag{5.40}$$

with $d_p^*(\mathbf{v}, \mathbf{w}) = \|\mathbf{v} - \mathbf{w}\|_p^p$. In general l_p-kernels can be defined by

$$K(\mathbf{v}, \mathbf{w}) = e^{-\frac{d_p(\mathbf{v},\mathbf{w})}{\sigma^p}}. \tag{5.41}$$

Obviously, in (5.41) $d_p(\mathbf{v}, \mathbf{w})$ can also be replaced by the parametrized counterparts $d_{p,\boldsymbol{\lambda}}(\mathbf{v}, \mathbf{w})$ and $d_{p,\boldsymbol{\Omega}}(\mathbf{v}, \mathbf{w})$. Thus, the gradients with respect to $\mathbf{w}$ follow immediately from the previous considerations in a trivial manner.

The subsections are based on

*A. Bohnsack, K. Domaschke, M. Kaden, M. Lange, T. Villmann, "*Learning Matrix Quantization and Relevance Learning Based on Schatten-*p*-norms*", Neurocomputing 192 (2016), pp. 104-114.[19]*

K. Domaschke, M. Kaden, M. Lange, T. Villmann, "Learning Matrix Quantization and Variants of Relevance Learning", *in M. Verleysen, ed., Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2015), pp. 13-18, Louvain-La-Neuve, Belgium (2015) [32].*

*M. Kaden, M. Lange, D. Nebel, M. Riedel and T. Geweniger and T. Villmann, "*Aspects in Classification Learning - Review of Recent Developments in Learning Vector Quantization*", Foundations of Computing and Decision Sciences 39 (2014), pp. 79-105.[64]*

*A. Villmann, M. Lange-Geisler, T. Villmann, "*About Semi-Inner Products for $\mathbf{p-QR}$-Matrix Norms*", TechReport (2018).[136]*

5.2 Learning Based on Matrix Norms

In this section, an extension of the LVQ approach to classify matrix data is proposed. The resulting learning matrix quantization (LMQ) algorithm is similar to LVQ but based on matrix norms. As mentioned before, Schatten-p-norms as generalization of l_p-norms are considered. Due to the use of matrix norms LMQ offers a greater structural flexibility compared to the vectorial counterpart. Several kinds of algebraic relevance weighting are introduced and their mathematical properties are discussed.

5.2.1 Learning Matrix Quantization (LMQ)

In LMQ the data are considered to be matrices $\mathbf{V} \in V_{n,m} \subseteq \mathbb{R}^{n \times m}$ and a set of prototypes W also given as matrices $\mathbf{W} \in \mathbb{R}^{n \times m}$. As already stated in section 3.6, the matrices generate a vector space $\mathcal{B}_{n,m}$ such that LMQ can be treated similarly to LVQ. Thus, all considerations regarding LMQ are consistent with the vectorial case, i. e. $m = 1$. In this sense, LMQ appears as the natural extension of LVQ.

Yet in LMQ, the Schatten-p-norm is used as dissimilarity measure, which represents an appropriate counterpart of GLVQ based on l_p-norms. This approach is denoted as *generalized learning matrix quantization* (GLMQ). The respective dissimilarity measure is

$$d_{s_p}(\mathbf{V}, \mathbf{W}) = \left(\|\mathbf{V} - \mathbf{W}\|_{S_p} \right)^p, \tag{5.42}$$

which is comparable to $d_p(\mathbf{v}, \mathbf{w})$ in GLVQ. The respective cost function of GLMQ

becomes

$$E_{GLMQ} = \frac{1}{2} \sum_{\mathbf{V} \in V_{m,n}} f\left(\mu_{s_p}(\mathbf{V})\right), \tag{5.43}$$

with the adjusted matrix-based classifier function

$$\mu_{s_p}(\mathbf{V}) = \frac{d^+_{s_p}(\mathbf{V}) - d^-_{s_p}(\mathbf{V})}{d^+_{s_p}(\mathbf{V}) + d^-_{s_p}(\mathbf{V})} \in [-1, 1]. \tag{5.44}$$

Likewise, formal prototype updates in GLMQ can be obtained by applying the analog formalism to the derivates as for GLVQ such that

$$\mathbf{W}^{\pm} \leftarrow \mathbf{W}^{\pm} + \eta_{\mathbf{W}} \cdot \triangle \mathbf{W}^{\pm}$$

with

$$\triangle \mathbf{W}^{\pm} = \frac{\partial f}{\partial \mu_{s_p}(\mathbf{V})} \cdot \frac{\partial \mu_{s_p}(\mathbf{V})}{\partial d^{\pm}_{s_p}(\mathbf{V})} \cdot \frac{\partial d^{\pm}_{s_p}(\mathbf{V})}{\partial \mathbf{W}^{\pm}}, \tag{5.45}$$

where $0 < \eta_{\mathbf{W}} \ll 1$ is also a decreasing learning rate like in vectorial case. The Frobenius norm with $p = 2$ is the most interesting case. For that the derivative $\frac{\partial d^{\pm}_{s_p}(\mathbf{V})}{\partial \mathbf{W}^{\pm}}$ in (5.45) becomes simply

$$\frac{\partial d^{\pm}_{s_2}(\mathbf{V})}{\partial \mathbf{W}^{\pm}} = -2\left(\mathbf{V} - \mathbf{W}^{\pm}\right). \tag{5.46}$$

Note that, the spatial information within the matrices is not taken into account when applying the Frobenius norm. The more general Schatten-p-norms have been proven to be successful in classification learning of images and time-resolved spectra [38, 32].

5.2.2 Relevance Learning in GLMQ

Several variants of relevance learning for GLMQ are introduced and discussed in this subsection. The algebraic structure of the Banach space $\mathcal{B}_{n,m}$ of matrices equipped with the Schatten-p-norm is more complex as the simpler $\mathbb{R}^n$. In detail, depending on the used algebraic composition there are several possibilities to define kinds of relevance learning in GLMQ. Matrix compositions with a relevance matrix $\mathbf{R}$ and tensor composites with a relevance tensor $\mathbf{T}$ are investigated, where the case of applying the Frobenius norm is always of particular interest. The following approaches are referred to the generic term of *relevance generalized learning matrix quantization* (GRLMQ), which are schematically summarized in Figure 5.5.

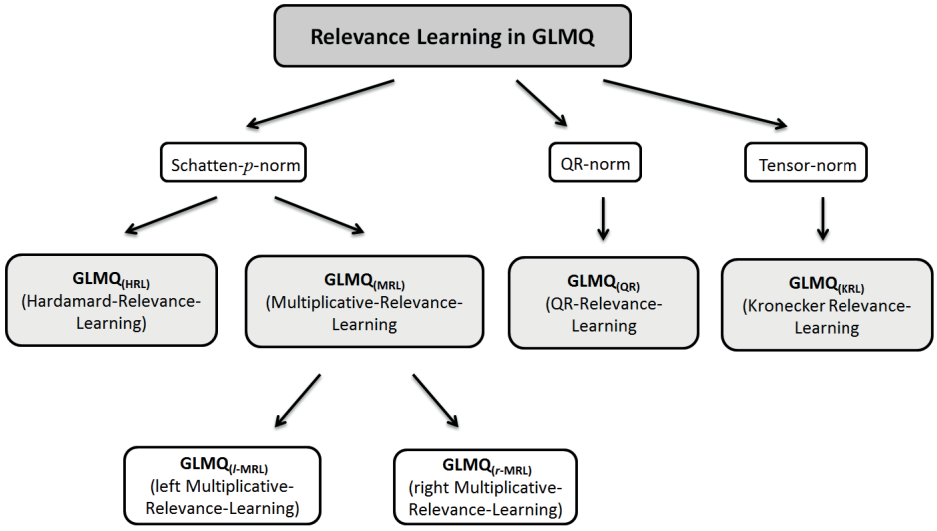

Figure 5.5: Schematic overview of the considered relevance types in GRLMQ.

Hadamard-Relevance-Learning (HRL)

Hadamard-Relevance-Learning in GLMQ (GLMQ_{HRL}) represents the counterpart of GRLVQ, introduced in subsection 2.5.2 on page 28. In GLMQ_{HRL} the dissimilarity measure

$$d_{s_p,\mathbf{R}\circ}(\mathbf{V},\mathbf{W}) = \left(\|\mathbf{R}\circ(\mathbf{V}-\mathbf{W})\|_{S_p}\right)^p \tag{5.47}$$

is taken with the relevance matrix $\mathbf{R}\in\mathbb{R}^{n\times m}$ independently wighting each entry of a data matrix $\mathbf{V}$ by applying of the Hadamard product $\circ$. The stochastic derivatives of E_{GLMQ} (5.43) with respect to $\mathbf{R}$ yield the relevance update

$$\mathbf{R}\leftarrow\mathbf{R}-\eta_{\mathbf{R}}\cdot\triangle\mathbf{R}$$

with

$$\triangle\mathbf{R} = -\frac{\partial f}{\partial\mu_{s_p}(\mathbf{V})}\cdot\left(\frac{\partial\mu_{s_p}(\mathbf{V})}{\partial d^+_{s_p,\mathbf{R}\circ}(\mathbf{V})}\cdot\frac{\partial d^+_{s_p,\mathbf{R}\circ}(\mathbf{V})}{\partial\mathbf{R}}+\frac{\partial\mu_{s_p}(\mathbf{V})}{\partial d^-_{s_p,\mathbf{R}\circ}(\mathbf{V})}\cdot\frac{\partial d^-_{s_p,\mathbf{R}\circ}(\mathbf{V})}{\partial\mathbf{R}}\right), \tag{5.48}$$

$0 < \eta_{\mathbf{R}} \ll 1$ is the learning rate for $\mathbf{R}$. In case of $p = 2$ the dissimilarity measure $d_{s_p,\mathbf{R}\circ}(\mathbf{V},\mathbf{W})$ and the formal derivative $\frac{\partial d^{\pm}_{s_p,\mathbf{R}\circ}(\mathbf{V})}{\partial \mathbf{R}}$ are reduced to

$$d_{s_2,\mathbf{R}\circ}(\mathbf{V},\mathbf{W}) = \mathsf{tr}\left((\mathbf{R}\circ(\mathbf{V}-\mathbf{W}))(\mathbf{R}\circ(\mathbf{V}-\mathbf{W}))^{\top}\right) \tag{5.49}$$

and

$$\frac{\partial d^{\pm}_{s_2,\mathbf{R}\circ}(\mathbf{V})}{\partial \mathbf{R}} = 2\mathbf{R}\circ\left(\mathbf{V}-\mathbf{W}^{\pm}\right)\circ\left(\mathbf{V}-\mathbf{W}^{\pm}\right), \tag{5.50}$$

respectively. The corresponding prototype update in $\mathrm{GLMQ}_{\mathrm{HRL}}$ implies the term

$$\frac{\partial d^{\pm}_{s_2,\mathbf{R}\circ}(\mathbf{V})}{\partial \mathbf{W}^{\pm}} = -2\mathbf{R}\circ\mathbf{R}\circ\left(\mathbf{V}-\mathbf{W}^{\pm}\right), \tag{5.51}$$

which is obtained from (5.45) for $p = 2$. Yet, the use of the Frobenius norm and the Hadamard product in the dissimilarity measure leads to the fact that the spatial matrix relations are not taken into account. Thus $\mathrm{GLMQ}_{\mathrm{HRL}}$ has no essential benefit over GRLVQ.

Multiplicative-Relevance-Learning (MRL)

A further kind of relevance learning results by applying the ordinary matrix multiplication instead of the Hadamard product. A differentiation is made between left- and right weighting by

$$d^{l}_{s_p,\mathbf{R}}(\mathbf{V},\mathbf{W}) = \left(\|\mathbf{R}\cdot(\mathbf{V}-\mathbf{W})\|_{S_p}\right)^{p} \tag{5.52}$$

$$d^{r}_{s_p,\mathbf{R}}(\mathbf{V},\mathbf{W}) = \left(\|(\mathbf{V}-\mathbf{W})\cdot\mathbf{R}\|_{S_p}\right)^{p}. \tag{5.53}$$

In (5.52) and (5.53) a weighted linear relevance mixing with $\mathbf{R}\in\mathbb{R}^{k\times n}$ (left case) or $\mathbf{R}\in\mathbb{R}^{m\times k}$ (right case) is applied, respectively. For $p = 2$ the dissimilarity measures reads as $d^{l}_{s_2,\mathbf{R}}(\mathbf{V},\mathbf{W}) = \mathrm{tr}\left((\mathbf{RA})\cdot(\mathbf{RA})^{\top}\right)$ or $d^{r}_{s_2,\mathbf{R}}(\mathbf{V},\mathbf{W}) = \mathrm{tr}\left((\mathbf{AR})\cdot(\mathbf{AR})^{\top}\right)$. The relevance updates can be obtained analogously to (5.48), i. e.

$$\frac{\partial d^{l,\pm}_{s_2,\mathbf{R}}(\mathbf{V})}{\partial \mathbf{R}} = 2\mathbf{R}\cdot\left(\mathbf{V}-\mathbf{W}^{\pm}\right)\cdot\left(\mathbf{V}-\mathbf{W}^{\pm}\right)^{\top} \tag{5.54}$$

for the left case and

$$\frac{\partial d^{r,\pm}_{s_2,\mathbf{R}}(\mathbf{V})}{\partial \mathbf{R}} = 2\left(\mathbf{V}-\mathbf{W}^{\pm}\right)^{\top}\cdot\left(\mathbf{V}-\mathbf{W}^{\pm}\right)\cdot\mathbf{R} \tag{5.55}$$

for the right case. The respective prototype adaptation include the derivatives

$$\frac{\partial d^{l,\pm}_{s_2,\mathbf{R}}(\mathbf{V})}{\partial \mathbf{W}^{\pm}} = -2\left(\mathbf{R}^{\top}\cdot\mathbf{R}\right)\cdot\left(\mathbf{V}-\mathbf{W}^{\pm}\right) \tag{5.56}$$

and

$$\frac{\partial d^{l,\pm}_{s_2,\mathbf{R}}(\mathbf{V})}{\partial \mathbf{W}^{\pm}} = -2\left(\mathbf{V}-\mathbf{W}^{\pm}\right)\cdot\left(\mathbf{R}\cdot\mathbf{R}^{\top}\right), \tag{5.57}$$

accordingly for the left and right variant. These methods are referred to as *left/right Multiplicative Relevance Learning* in GLMQ ($\text{GLMQ}_{\text{l/r-MRL}}$). A noticeable benefit of this approach is that now correlation between matrix entries are taken into account also for the case $p = 2$. More precisely, $\text{GLMQ}_{\text{l-MRL}}$ relates to correlation of the columns and $\text{GLMQ}_{\text{r-MRL}}$ to correlation of the rows. Thus, Frobenius norm together with $\text{GLMQ}_{\text{l/r-MRL}}$ treats the data matrix entries partially dependent, see second row in Figure 5.6.

QR-Relevance-Learning (QR)

The simultaneous learning of correlations for row and column dependencies of the matrix entries leads to the application of the **QR**-norm in GLMQ. This method is denoted as **QR**-Relevance-Learning in GLMQ (GLMQ_{QR}). In subsection 3.6.3 the **QR**-norm $\|\mathbf{A}\|_{\mathbf{Q},\mathbf{R}} = \sqrt{\text{tr}\left(\mathbf{Q}\mathbf{A}\mathbf{R}^{\top}\mathbf{A}^{\top}\right)}$ was introduced, where $\mathbf{Q}\in\mathbb{R}^{n\times n}$ and $\mathbf{R}\in\mathbb{R}^{m\times m}$. For **QR**-Relevance-Learning the dissimilarity is defined by

$$d_{\mathbf{QR}}(\mathbf{V},\mathbf{W}) = \text{tr}\left(\mathbf{Q}\cdot(\mathbf{V}-\mathbf{W})\,\mathbf{R}^{\top}(\mathbf{V}-\mathbf{W})^{\top}\right). \tag{5.58}$$

and the derivative with respect to the prototype becomes

$$\frac{\partial d^{\pm}_{\mathbf{QR}}(\mathbf{V})}{\partial \mathbf{W}^{\pm}} = -\mathbf{Q}^{\top}\left(\mathbf{V}-\mathbf{W}^{\pm}\right)\mathbf{R}^{\top} - \mathbf{Q}\left(\mathbf{V}-\mathbf{W}^{\pm}\right)\mathbf{R}. \tag{5.59}$$

$\mathbf{Q}$ and $\mathbf{R}$ are assumed to be positive definite matrices to ensure the norm properties. In order to satisfy this the decompositions $\mathbf{Q} = \widetilde{\mathbf{Q}}^{\top}\widetilde{\mathbf{Q}}$ and $\mathbf{R} = \widetilde{\mathbf{R}}\widetilde{\mathbf{R}}^{\top}$ are supposed with arbitrary matrices $\widetilde{\mathbf{Q}}\in\mathbb{R}^{n\times n}$ and $\widetilde{\mathbf{R}}\in\mathbb{R}^{m\times m}$. Beside the prototypes the relevance matrices $\widetilde{\mathbf{Q}}$ and $\widetilde{\mathbf{R}}$ are adapted simultaneously according to

$$\widetilde{\mathbf{Q}} \leftarrow \widetilde{\mathbf{Q}} - \eta_{\widetilde{\mathbf{Q}}}\cdot\triangle\widetilde{\mathbf{Q}} \tag{5.60}$$

$$\widetilde{\mathbf{R}} \leftarrow \widetilde{\mathbf{R}} - \eta_{\widetilde{\mathbf{R}}}\cdot\triangle\widetilde{\mathbf{R}} \tag{5.61}$$

with

$$\triangle\widetilde{\mathbf{Q}} = -\frac{\partial f}{\partial \mu_{s_p}(\mathbf{V})} \cdot \left(\frac{\partial \mu_{s_p}(\mathbf{V})}{\partial d^{+}_{\mathbf{QR}}(\mathbf{V})} \cdot \frac{\partial d^{+}_{\mathbf{QR}}(\mathbf{V})}{\partial \widetilde{\mathbf{Q}}} + \frac{\partial \mu_{s_p}(\mathbf{V})}{\partial d^{-}_{\mathbf{QR}}(\mathbf{V})} \cdot \frac{\partial d^{-}_{\mathbf{QR}}(\mathbf{V})}{\partial \widetilde{\mathbf{Q}}} \right) \tag{5.62}$$

$$\triangle\widetilde{\mathbf{R}} = -\frac{\partial f}{\partial \mu_{s_p}(\mathbf{V})} \cdot \left(\frac{\partial \mu_{s_p}(\mathbf{V})}{\partial d^{+}_{\mathbf{QR}}(\mathbf{V})} \cdot \frac{\partial d^{+}_{\mathbf{QR}}(\mathbf{V})}{\partial \widetilde{\mathbf{R}}} + \frac{\partial \mu_{s_p}(\mathbf{V})}{\partial d^{-}_{\mathbf{QR}}(\mathbf{V})} \cdot \frac{\partial d^{-}_{\mathbf{QR}}(\mathbf{V})}{\partial \widetilde{\mathbf{R}}} \right) \tag{5.63}$$

and

$$\frac{\partial d^{\pm}_{\mathbf{QR}}(\mathbf{V})}{\partial \widetilde{\mathbf{Q}}} = \widetilde{\mathbf{Q}} \left(\mathbf{V} - \mathbf{W}^{\pm}\right)^{\top} \mathbf{R} \left(\mathbf{V} - \mathbf{W}^{\pm}\right)^{\top} + \widetilde{\mathbf{Q}} \left(\mathbf{V} - \mathbf{W}^{\pm}\right) \mathbf{R} \left(\mathbf{V} - \mathbf{W}^{\pm}\right)^{\top} \tag{5.64}$$

$$\frac{\partial d^{\pm}_{\mathbf{QR}}(\mathbf{V})}{\partial \widetilde{\mathbf{R}}} = \left(\mathbf{V} - \mathbf{W}^{\pm}\right)^{\top} \mathbf{Q} \left(\mathbf{V} - \mathbf{W}^{\pm}\right) \widetilde{\mathbf{R}} + \left(\mathbf{V} - \mathbf{W}^{\pm}\right) \mathbf{Q}^{\top} \left(\mathbf{V} - \mathbf{W}^{\pm}\right)^{\top} \widetilde{\mathbf{R}}. \tag{5.65}$$

The learning rates $0 < \eta_{\widetilde{\mathbf{Q}}} \ll 1$ and $0 < \eta_{\widetilde{\mathbf{R}}} \ll 1$ can be initialized independent from each other.

The obvious advantage of $\mathrm{GLMQ}_{\mathrm{QR}}$ compared $\mathrm{GLMQ}_{\mathrm{l/r\text{-}MRL}}$ is that $\mathrm{GLMQ}_{\mathrm{QR}}$ allows to consider all correlations between rows and columns of the data matrices simultaneously, where $\widetilde{\mathbf{Q}}$ is mainly sensitive to column relations and $\widetilde{\mathbf{R}}$ to row relations. Hence, both matrices play a similar role like the classification correlation matrix $\mathbf{\Lambda}$ in GMLVQ due to

$$\begin{aligned} \mathrm{tr}\left(\mathbf{Q}\left(\mathbf{V} - \mathbf{W}\right)\mathbf{R}^{\top}\left(\mathbf{V} - \mathbf{W}\right)^{\top}\right) &= \mathrm{tr}\left(\widetilde{\mathbf{Q}}^{T}\widetilde{\mathbf{Q}} \cdot \left(\mathbf{V} - \mathbf{W}\right)\mathbf{R}^{\top}\left(\mathbf{V} - \mathbf{W}\right)^{\top}\right) \\ &= \mathrm{tr}\left(\left(\mathbf{V} - \mathbf{W}\right)^{\top} \cdot \widetilde{\mathbf{Q}}^{T}\widetilde{\mathbf{Q}} \cdot \left(\mathbf{V} - \mathbf{W}\right)\mathbf{R}^{\top}\right) \\ &= \mathrm{tr}\left(\left(\widetilde{\mathbf{Q}} \cdot \left(\mathbf{V} - \mathbf{W}\right)\right)^{2}\mathbf{R}^{\top}\right) \end{aligned}$$

and

$$\begin{aligned} \mathrm{tr}\left(\mathbf{Q}\left(\mathbf{V} - \mathbf{W}\right)\mathbf{R}^{\top}\left(\mathbf{V} - \mathbf{W}\right)^{\top}\right) &= \mathrm{tr}\left(\mathbf{Q}\left(\mathbf{V} - \mathbf{W}\right)\widetilde{\mathbf{R}}\widetilde{\mathbf{R}}^{\top}\left(\mathbf{V} - \mathbf{W}\right)^{\top}\right) \\ &= \mathrm{tr}\left(\mathbf{Q}\left(\left(\mathbf{V} - \mathbf{W}\right)\widetilde{\mathbf{R}}\right)^{2}\right), \end{aligned}$$

respectively. Note that, $\mathrm{GLMQ}_{\mathrm{QR}}$ is no counterpart to GMLVQ, because the entries of $\mathbf{\Lambda}$ correspond to correlations of all features of data vectors and $\mathbf{Q}$ ($\mathbf{R}$) consider only correlations of columns (rows) of data matrices, see Figure 5.6. Therefore, $\mathbf{Q}$

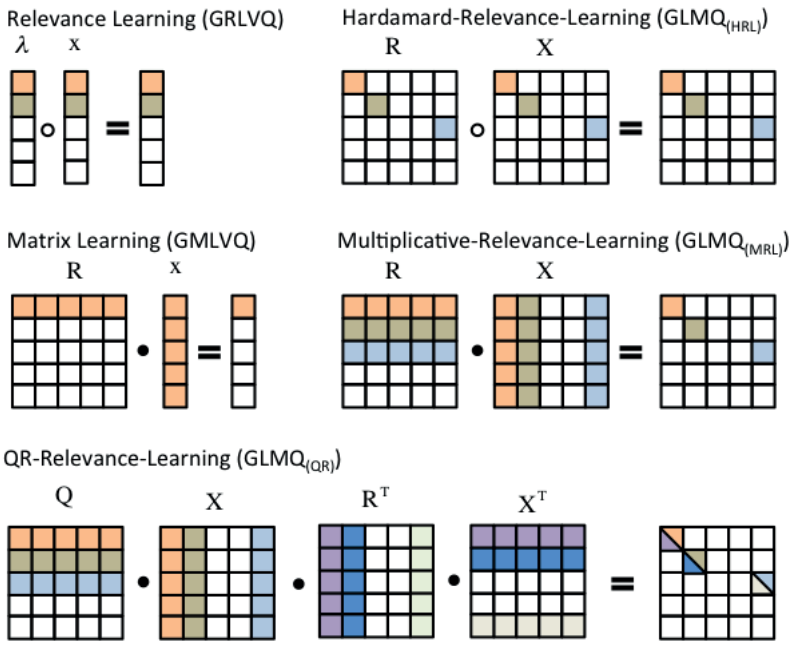

Figure 5.6: Schematic representation of several kinds of relevance learning for GLVQ and GLMQ. The first line shows the comparison of GRLVQ and $\mathrm{GLMQ}_{\mathrm{HRL}}$, where it can be seen that the spatial relations are not taken into account. In contrast, the second rows shows that the the spatial relations are taken into account by applying $\mathrm{GLMQ}_{\mathrm{l/r\text{-}MRL}}$, but it is no counterpart to GMLVQ. The last row shows how the spatial relations are taken into account by the simultaneous use of the relevance matrices $\mathbf{Q}$ and $\mathbf{R}$ by applying the $\mathbf{QR}$-norm.

and $\mathbf{R}$ represent not a classification correlation matrix in the strict sense. The relation matrices of $\mathrm{GLMQ}_{\mathrm{l/r\text{-}MRL}}$ can be interpreted similarly:
Because of

$$\begin{aligned}\mathrm{tr}\left((\mathbf{R}(\mathbf{V}-\mathbf{W}))\cdot(\mathbf{R}(\mathbf{V}-\mathbf{W}))^\top\right) &= \mathrm{tr}\left(\mathbf{R}(\mathbf{V}-\mathbf{W})(\mathbf{V}-\mathbf{W})^\top\mathbf{R}^\top\right)\\ &= \mathrm{tr}\left((\mathbf{V}-\mathbf{W})^\top\mathbf{R}^\top\mathbf{R}(\mathbf{V}-\mathbf{W})\right)\\ &= \mathrm{tr}\left((\mathbf{V}-\mathbf{W})^\top\mathbf{\Lambda}_{\mathrm{left}}(\mathbf{V}-\mathbf{W})\right)\\ &= \mathrm{tr}\left((\mathbf{R}(\mathbf{V}-\mathbf{W}))^2\right)\end{aligned}$$

and

$$\begin{aligned}\mathrm{tr}\left(((\mathbf{V}-\mathbf{W})\mathbf{R})\cdot((\mathbf{V}-\mathbf{W})\mathbf{R})^\top\right) &= \mathrm{tr}\left((\mathbf{V}-\mathbf{W})\mathbf{R}\cdot\mathbf{R}^\top(\mathbf{V}-\mathbf{W})^\top\right)\\ &= \mathrm{tr}\left((\mathbf{V}-\mathbf{W})\mathbf{\Lambda}_{\mathrm{right}}(\mathbf{V}-\mathbf{W})^\top\right)\\ &= \mathrm{tr}\left(((\mathbf{V}-\mathbf{W})\mathbf{R})^2\right)\end{aligned}$$

the matrices $\mathbf{\Lambda}_{\mathrm{left}}$ and $\mathbf{\Lambda}_{\mathrm{right}}$ can be interpreted as a kind of classification correlation matrix, which consists only of correlations of columns and rows of data matrices, respectively.

Further, $\mathrm{GLMQ}_{\mathrm{QR}}$ reduces to the common LMQ approach when choosing $\mathbf{Q}=\mathbf{I}$ and $\mathbf{R}=\mathbf{I}$. Note that, for $\mathbf{Q}=\mathbf{I}$ or $\mathbf{R}=\mathbf{I}$ the $\mathbf{QR}$-norm results in $\|\mathbf{A}\|_{\mathbf{I},\mathbf{R}} = \sqrt{\mathrm{tr}\left(\mathbf{A}\mathbf{R}^\top\mathbf{A}^\top\right)}$ or
$\|\mathbf{A}\|_{\mathbf{Q},\mathbf{I}} = \sqrt{\mathrm{tr}\left(\mathbf{Q}\mathbf{A}\mathbf{A}^\top\right)}$, respectively, which differs from $\|\mathbf{A}\mathbf{R}\|_{\mathcal{S}_2} = \sqrt{\mathrm{tr}\left((\mathbf{A}\mathbf{R})\cdot(\mathbf{A}\mathbf{R})^\top\right)}$ and $\|\mathbf{Q}\mathbf{A}\|_{\mathcal{S}_2} = \sqrt{\mathrm{tr}\left((\mathbf{Q}\mathbf{A})\cdot(\mathbf{Q}\mathbf{A})^\top\right)}$. Therefore, $\mathrm{GLMQ}_{\mathrm{QR}}$ does not exactly correspond to the combination of left/right Multiplicative Relevance Learning.

Kronecker-Relevance-Learning (KRL)

Another algebraic kind of multiplication for matrices can be obtained via the Kronecker-product $\mathbf{K}=\mathbf{A}\otimes\mathbf{B}$ between matrices $\mathbf{A}\in\mathbb{R}^{m_1\times n_1}$ and $\mathbf{B}\in\mathbb{R}^{m_2\times n_2}$, which is considered for relevance learning in GLMQ [36, 86]. It is known from tensor algebra that the outcomes of the Kronecker-product have a huge dimensionality, i. e. $\mathbf{K}\in\mathbb{R}^{m_1m_2\times n_1n_2}$. Thus, a respective relevance composition implies a high variability to relate the matrix elements having in mind the idea of high-dimensional embedding

as known from SVMs [121].
The dissimilarity measure for *Kronecker-Relevance-Learning* in GLMQ ($\mathrm{GLMQ}_{\mathrm{KRL}}$) reads as

$$d_{s_p,\mathbf{R}\otimes}(\mathbf{V},\mathbf{W}) = \left(\|\mathbf{R}\otimes(\mathbf{V}-\mathbf{W})\|_{S_p}\right)^p \tag{5.66}$$

$$= \mathrm{tr}\left(|\mathbf{R}\otimes(\mathbf{V}-\mathbf{W})|^p\right). \tag{5.67}$$

Because of $|\mathbf{A}| = \sqrt[+]{\mathbf{A}^*\mathbf{A}}$ (the positive matrix root) the dissimilarity measure $d_{s_p,\mathbf{R}\otimes}(\mathbf{V},\mathbf{W})$ in (5.67) can be simplified to

$$d_{s_p,\mathbf{R}\otimes}(\mathbf{V},\mathbf{W}) = \mathrm{tr}\left(\left(\sqrt[+]{\mathbf{R}^\top\mathbf{R}\otimes(\mathbf{V}-\mathbf{W})^\top(\mathbf{V}-\mathbf{W})}\right)^p\right),$$

which reduces to

$$d_{s_{2q},\mathbf{R}\otimes}(\mathbf{V},\mathbf{W}) = \mathrm{tr}\left(\left(\mathbf{R}^\top\mathbf{R}\otimes(\mathbf{V}-\mathbf{W})^\top(\mathbf{V}-\mathbf{W})\right)^q\right)$$

for $p = 2q$. For the most interesting case $p = 2$ the dissimilarity measure becomes

$$d_{s_2,\mathbf{R}\otimes}(\mathbf{V},\mathbf{W}) = \mathrm{tr}\left(\left(\mathbf{R}^\top\mathbf{R}\otimes(\mathbf{V}-\mathbf{W})^\top(\mathbf{V}-\mathbf{W})\right)\right) \tag{5.68}$$

$$= \mathrm{tr}\left(\mathbf{R}^\top\mathbf{R}\right)\mathrm{tr}\left(\left((\mathbf{V}-\mathbf{W})^\top(\mathbf{V}-\mathbf{W})\right)\right), \tag{5.69}$$

where the commutative properties of the Kronecker-product in the last equation (5.69) is applied [86]. According to the last transformation, it can be concluded that the structure of the relevance matrix $\mathbf{R}$ is ignored in the dissimilarity measure despite the trace value in case of $p = 2$. Hence, relevance learning in $\mathrm{GLMQ}_{\mathrm{KRL}}$ is meaningless due to the expected variability, which is degenerated to one parameter only in this context.

Tensor-Relevance-Learning

A further kind of algebraic composition are tensors as multi-linear maps applied to matrices. For example, covariance structures of data sets consisting of matrices can be described by tensors, such that an appropriate PCA can be performed [18, 59, 83]. Mathematically, matrices can be seen as tensors of second order and vectors as tensors of first order. Regarding that, the mapping $\boldsymbol{\Omega}(\mathbf{v}-\mathbf{w})$ used in GMLVQ can also be defined by a tensor composite $\overset{\langle 2\rangle}{\boldsymbol{\Omega}}[\mathbf{v}-\mathbf{w}]$, which results in a tensor of first order

(vector). Now, this idea is transferred to the dissimilarities of GLMQ.

Relevance Tensors of Fourth Order To define a dissimilarity measure of tensor-relevance-learning of fourth order the tensor induced mapping

$$\overset{\langle 4\rangle}{\mathbf{T}}[\mathbf{V}-\mathbf{W}]=\mathbf{X}$$

is considered, where the matrix difference $\mathbf{V}-\mathbf{W}$ is projected onto a matrix $\mathbf{X}$ by $X_{ij}=\sum_{k,l}T_{ijkl}\left(V_{kl}-W_{kl}\right)$. Based on Schatten-$p$-norms the dissimilarity measure reads as

$$d_{s_p,\overset{\langle 4\rangle}{\mathbf{T}}[\bullet]}(\mathbf{V},\mathbf{W})=\left(\left\|\overset{\langle 4\rangle}{\mathbf{T}}[\mathbf{V}-\mathbf{W}]\right\|_{S_p}\right)^p.$$

However, $\mathbf{V}-\mathbf{W}$ as well as $\mathbf{X}$ are elements of the matrix vector spaces describing linear mappings as known from linear algebra. For those tensor compositions, a base representation in these vector spaces can always be generated such that the tensor composition is equivalent to a composition of the Kronecker-product [36, 86]. Hence, the respective relevance learning variability is the same as for the previously introduced relevance learning in $\mathrm{GLMQ}_{\mathrm{KRL}}$. In detail, for the most interesting case $p=2$ it is also algebraically degenerated.

Relevance Tensors of Third Order As last variant of relevance learning for GLMQ tensors of third order are investigated. The respective mapping

$$\overset{\langle 3\rangle}{\mathbf{T}}[\mathbf{V}-\mathbf{W}]=\mathbf{t} \tag{5.70}$$

results in a vector $\mathbf{t}$ with $t_i=\sum_{jk}T_{ijk}\left(V_{jk}-W_{jk}\right)$. By applying the mapping (5.70) the Schatten-p-norm is simplified to the l_p-norm. Thus, the respective dissimilarity measure looks as follows

$$d_{s_p,\overset{\langle 3\rangle}{\mathbf{T}}[\bullet]}(\mathbf{V},\mathbf{W})=\left(\left\|\overset{\langle 3\rangle}{\mathbf{T}}[\mathbf{V}-\mathbf{W}]\right\|_{l_p}\right)^p. \tag{5.71}$$

The derivative of (5.71) can be easily calculated. At first glance, relevance learning GLMQ with tensors of third order seems to be a counterpart of GMLVQ, which applies to the matrix mapping $\mathbf{\Omega}\left(\mathbf{v}-\mathbf{w}\right)$. Unfortunately, this is not the case because the dimensionality of the result of the tensor operation is a vector (tensor of zero order),

whereas the difference $\mathbf{V} - \mathbf{W}$ is a tensor of second order. In contrast, the mapping in GMLVQ retains the orders of the tensors.

5.3 Numerical Simulations and Selected Applications

The first part of this subsection shows and compares the use of the weighted l_1-distance with the weighted Euclidean distance in GLVQ for two applications, where also a comparison to an earlier but inconsistent LVQ approach for the first application is done. It will be shown that l_1-distances can be successfully applied by using the above described approximation techniques for l_p-norms, which generate a consistent approach with the l_1-norm and improves the classification performance.

Subsequently, the LMQ algorithm based on Schatten-p-norms for $p = 2$ together with possibilities of relevance learning is applied on two different matrix data sets. For comparison to the vectorial case, also GLVQ, GRLVQ and GMLVQ are applied to the vectorized version of given matrix data sets. These exemplary numerical simulations demonstrate the benefit of using the matrix counterpart of LVQ to classify matrix data.

The subsection is based on

M. Lange, D. Zühlke, O. Holz, T. Villmann, "Applications of l_p-norms and their Smooth Approximations for Gradient Based Learning Vector Quantization", *in M. Verleysen, ed., Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2014),pp. 271-276. [82]*

5.3.1 LVQ based on l_p-norms

Application in Tiling Microarray Data

Applying GMLVQ with the dissimilarity $d_{\infty,\mathbf{\Omega}}(\mathbf{v}, \mathbf{w})$ for two different datasets. The first dataset is denoted as "tiling microarray data", which is a set of 4120 tiling microarrays (with data dimension $n = 24$) corresponding to exonic and intronic/intragenic regions in chromosome 3 of C.elegans. Usually, learning of tiling micro array data is difficult because they contain a lot of noise. Training is based on the current annotation of the genome. A more detailed description of these data can be found in [13]. In this publication, the standard LVQ1 algorithm based on the (weighted) Euclidean

norm $d_{2,\boldsymbol{\lambda}}(\mathbf{v},\mathbf{w})$ was applied to learn the prototypes. However, inherent prototype selection as well as the optimization of the relevance parameters λ_i uses the l_1-distance $d_{1,\boldsymbol{\lambda}}(\mathbf{v},\mathbf{w})$ to handle noisy data. Thus, the used approach is inconsistent. In [13], the best classification performance with 89.3% was obtained for 6 prototypes per class.

Now, GMLVQ is applied with l_1-norm and l_2-norm with consistent derivatives. That means, the above introduced smooth approximations for the l_1-norm to learn the prototypes in GMLVQ are applied. For comparison to the mentioned publication [13], the same number of prototypes (6 per class) for both variants is used and matrix adaptation in a 4-fold cross-validation is performed. This results in a test accuracies of 90.8% for using the l_1-norm and 88.8% for the Euclidean variant. Therefore, the consistent approach with the l_1-norm improves the classification performance. Further, no significant difference regarding the used approximation function, α-*quasi-absolute* function $|x|_\alpha^{\mathcal{Q}}$ and α-*soft-absolute* function $|x|_\alpha^{S}$ with $\alpha = 20$, can be recognized.

Application in Gas Chromatography– Mass Spectrometry (GC-MS) Spectra

A second application considers spectral data. This data was obtained from a gas chromatography– mass spectrometry (GC-MS) analysis of volatile organic components in exhaled breath to detect inflammatory processes in the lung. The GC-MS spectra were delivered as 334-dimensional spectra covering a measurement time interval of 20 min. The dataset contains 48 spectra partitioned into two classes. A detailed explanation can be found in [33]. Again, GMLVQ with l_1-norm and l_2-norm is applied in a consistent way for prototype learning and matrix adaptation. However, here only one prototype per class with a 8-fold cross-validation is used. It results in a better test accuracy of 85.4% for the GMLVQ with l_1-norm than those for the Euclidean variant with 81.3%. This observation is in agreement with the knowledge about the noise influence for GC-MS spectra regarding the peak height, see [141], and also with the above previously stated robust behavior of l_p- distances for smaller p-values as reported in [38].

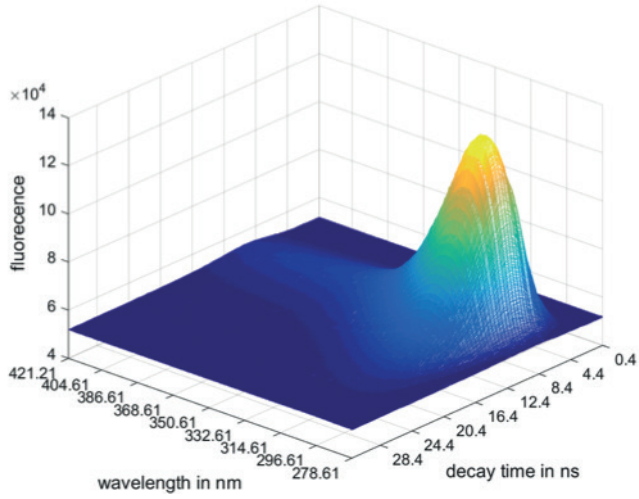

Figure 5.7: Two-dimensional signal (matrix) of fluorescence intensity at special emission energies and time points after fluorescence initialization.

The subsections are based on

A. Bohnsack, K. Domaschke, M. Kaden, M. Lange and T. Villmann, "Learning Matrix Quantization and Relevance Learning Based on Schatten-p-norms", *Neurocomputing 192 (2016), pp. 104-114.[19]*

5.3.2 LMQ based on Schatten-p-norms

In this subsection exemplary applications of the previously introduced GLMQ including also relevance learning are presented. In detail, several possibilities of the GLMQ approach regarding algebraic kinds of relevance learning are demonstrated and compared with relevance learning of GLVQ. As stated in the section of LMQ, the investigations are restricted to the most prominent case $p = 2$.

Application in Time Resolved Laser induced Fluorescence Spectroscopy

The Time Resolved Laser induced Fluorescence Spectroscopy (TRLFS) developed over the last 20 years is an important method to distinguish different substances or to characterize composite samples. During the measurements a laser beam stimulate

selected molecules of a desired substance which thereupon dissipates this energy thermally or optically in terms of auto-fluorescence. The latter leads to a time-dependent emission and can be measured as a two-dimensional signal (matrix). This signal represents the fluorescence intensity depending on the emission energy and the time after fluorescence initialization, visualized in Figure 5.7.Further, these signals reflect important substance specific characteristics in both dimensions to identify samples [78]. TRLF-spectra are functional data parametrized by the emission wavelength and the decay time, such that they can be handled as functional data depending on two variables. In general, the analysis of those spectra can be done by the adaption of mathematical data simulation models of given samples to estimate important parameters like the position of the emission maximum, the signal width or the decay of the fluorescence intensity [123, 129]. The estimated parameters are used for further analysis and investigation such as classification of real world spectra. However, because of poor model information and general serious numerical instabilities of the simulation models as well as poor confidential parameter estimations the respective considerations fail frequently. Hence, methods for the investigation of original spectra are required.

Simulated Spectra The fluorescence intensity $f_{em}(\omega)$ as a function of the emission wavelength ω for a single fluorophore is defined by

$$f_{em}(\omega) = Ae^{-\left(\frac{\omega-\mu}{\sigma}\right)^2}, \tag{5.72}$$

which is Gaussian or log-normal distributed [123]. For simplicity, the Gaussian shaped fluorescence intensities are used and only one fluorophore per class is assumed for simulations. In (5.72) the fluorescent substance specific parameters are the position of emission maximum μ and the width of the signal σ. Besides the wavelength dependent shape of fluorescence, the time resolved measurements provide also the decay behavior of fluorescence light $f_{tr}(\tau)$ over the time τ as additional information for the characterization of different substances. The signals shows how the intensity of the fluorescence light rises depending on the laser excitation as well as the camera system and, furthermore, how it decreases while emitting the received energy by autofluorescence. Note that, for real world data the excitation signal of the fluorescence rises in a Gaussian shape and, further, the emitting process follows an exponential performance, in general. The time dependent signal can be approximated at any

wavelength by folding these two functions [129]:

$$f_{tr}(\tau) = s\left(e^{-\frac{\tau - t_0}{\tau}} e^{\frac{s^2}{4\tau^2}} \left(1 + \mathsf{erf}\left(\frac{\tau - t_0}{s} - \frac{s}{2\tau_0}\right)\right)\right) \tag{5.73}$$

In (5.73) t_0 and s are fixed camera parameters whereas τ_0 is the substance specific decay of the fluorescence intensity and $\mathsf{erf}(x)$ is the Gauss error function[1]. By adjustment of this information the data matrices can be simulated by

$$\mathbf{F} = \mathbf{f}_{em} \cdot \mathbf{f}_{tr}^{'}, \tag{5.74}$$

where $\mathbf{f}_{em} = (f_{em}(\omega_1), \ldots, f_{em}(\omega_{N_\omega}))^\top$ and $\mathbf{f}_{tr} = (f_{tr}(\tau_1), \ldots, f_{tr}(\tau_{N_\omega}))^\top$ are the vectors of emission and time dependent intensities, respectively.
For the application of relevance learning LMQ two different data sets of simulated TRLF-spectra are created. They are denoted as TRLF1 and TRLF2. Both of them consist of two classes, where each is represented by 100 spectra of size 50×40 for emission wavelength ω and time τ, respectively. The two spectrum classes of TRLF1 differ in time decay, whereas the classes of TRLF2 can be distinguished according to the emission wavelength distribution, see Figure 5.8.Any other parameters were kept fixed in average yet taking into account underlying Gaussian noise. Further, it exists an additive class independent white noise on the whole data matrices. Thus, both data sets can be clearly distinguished by their classification characteristics regarding the focus on the dimension, such that the influence of several kinds of relevance learning for GLMQ can be investigated. To compare GLMQ-results with those of GLVQ, the data is vectorized by integration over time results in TRLF1-ω and TRLF2-ω. Analogously, the vectorized counterparts TRLF1-τ and TRLF2-τ are obtained by integration over the wavelength ω.

Real World Spectra The real world TRFL data set (R-TRFL) consists of TRFL-spectra of a single biological compound at two different excitation energy levels, see the last line in Figure 5.8. It is assumed that the substrate response and this behavior is reflected in the measured signal matrices [27]. Emphasizing the responses can be quite similar and in advance it is ambiguous, which dimension wavelength ω or time τ delivers the crucial information for class separation. Hence, matrix processing of the data is essential. In detail, for each of the two classes there are 60 data samples with a resolution of 100×20 for emission wavelength and time, respectively. Commonly, the

[1]The Gauss error function is defined as $\mathsf{erf}(x) = \frac{1}{\sqrt{\Pi}} \int_{-x}^{x} e^{-t^2} dt$.

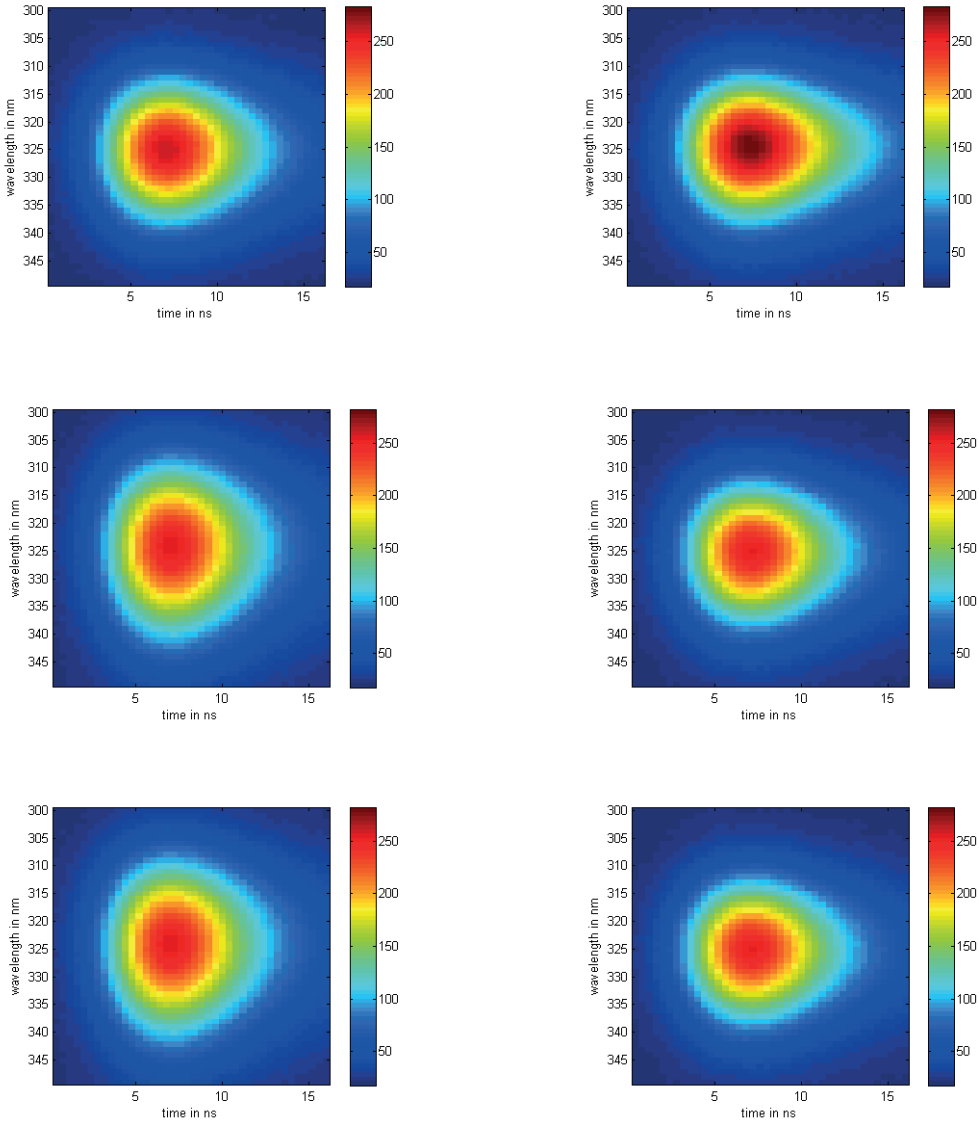

Figure 5.8: First line: Class means of the TRLF1 data set: The classes differ in the time decay (horizontal axes).
Second line: Class means of the TRLF2 data set. The classes differ with respect to the wavelength domain (vertical axes).
Last line: Class means of the R-TRLF data set. The classes differ according to the excitation energy levels.

spectra are considered irrespective of the time dependency. Thus, a respective data set R-TRFL-ω is generated from R-TRFL by integration over the time for comparison of GLMQ and GLVQ variants.

Simulation Results of TRLF spectra The simulated data and the real word data are investigated by the basis approach of GLMQ (without relevance learning) and GLMQ with several variants of relevance learning. A 10-fold cross-validation for R-TRLF data and a 5-fold cross-validation for the simulated data is applied for all experiments. Only one prototype per class and a constant learning rate $\eta = 0.01$ is used, where 1000 learning epochs for each run are applied. The resulting averaged test accuracies are summarized in Table 5.1 and the respective standard deviations are shown in Table 5.2.

data/method	GLMQ	GLMQ_{HRL}	$\text{GLMQ}_{\text{r-MRL}}$	$\text{GLMQ}_{\text{l-MRL}}$	GLMQ_{QR}
R-TRLF	83.3	81.0	80.4	82.4	93.1
TRLF1	74.5	76.5	78.0	79.0	81.0
TRLF2	88.0	89.0	93.5	90.5	94.5

Table 5.1: Averaged test accuracies in % for the TRLF data using different GLMQ variants without and with several relevance learning methods.

data/method	GLMQ	GLMQ_{HRL}	$\text{GLMQ}_{\text{r-MRL}}$	$\text{GLMQ}_{\text{l-MRL}}$	GLMQ_{QR}
R-TRLF	0.0910	0.108	0.099	0.1060	0.082
TRLF1	0.0779	0.0720	0.0570	0.0627	0.0675
TRLF2	0.0597	0.0418	0.0285	0.0371	0.0411

Table 5.2: Standard deviations related to the accuracies presented in Table 5.1.

According to these results, $\text{GLMQ}_{\text{r-MRL}}$ and $\text{GLMQ}_{\text{l-MRL}}$ lead to different results for TRLF1 and TRLF2 caused by their different class behavior regarding time or wavelength dependency. In detail, TRLF1 benefits by applying $\text{GLMQ}_{\text{l-MRL}}$ and for TRLF2 $\text{GLMQ}_{\text{r-MRL}}$ is more suitable. Further, for all three data sets the more complex GLMQ_{QR} yields the best results. The prototypes of an exemplary run for the R-TRLF data by using $\text{GLMQ}_{\text{l-MRL}}$, $\text{GLMQ}_{\text{r-MRL}}$, and GLMQ_{QR} are shown in Figure 5.10 and the respective relevance matrices for these methods are depicted in Figure 5.9. The comparison between the relevance matrices shows that $\mathbf{Q}$ and $\mathbf{R}$ contain more structural information than the left and right side relevance matrices

data/method	GLVQ	GRLVQ	GMLVQ
R-TRLF-ω	77.1	79.7	81.7
TRLF1-τ	76.0	77.0	81.0
TRLF1-ω	60.0	67.0	72.0
TRLF1-τ	80.0	84.5	87.0
TRLF1-ω	85.5	84.0	91.0

Table 5.3: Averaged test accuracies in % for the TRLF data applying different GLVQ variants without and with several relevance learning methods.

by $\text{GLMQ}_{\text{l/r-MRL}}$. Unfortunately, a satisfactory interpretation of the learned relevance matrices as for $\mathbf{\Lambda}$ in GLVQ is not possible, see explanations 5.2.2 on page 107. Especially for the R-TRLF data solely GLMQ_{QR} yields a considerable improvement in comparison to the basis approach of GLMQ without relevance learning, see Table 5.1. Further, also GLVQ is applied to the vectorized R-TRLF data. The obtained accuracies are listed in Table 5.3. According to them, the performance of GLVQ and GRLVQ is weaker, whereas GMLVQ is at least comparable to the GLMQ with simple relevance learning variants. Nevertheless, the performance of GLMQ_{QR} remains the best. This is probably due to the fact that the linear dependencies of the columns and rows within data matrices are learned simultaneously by GLMQ_{QR}. Similar classification performances are also obtained for the simulated data, see Table 5.3.

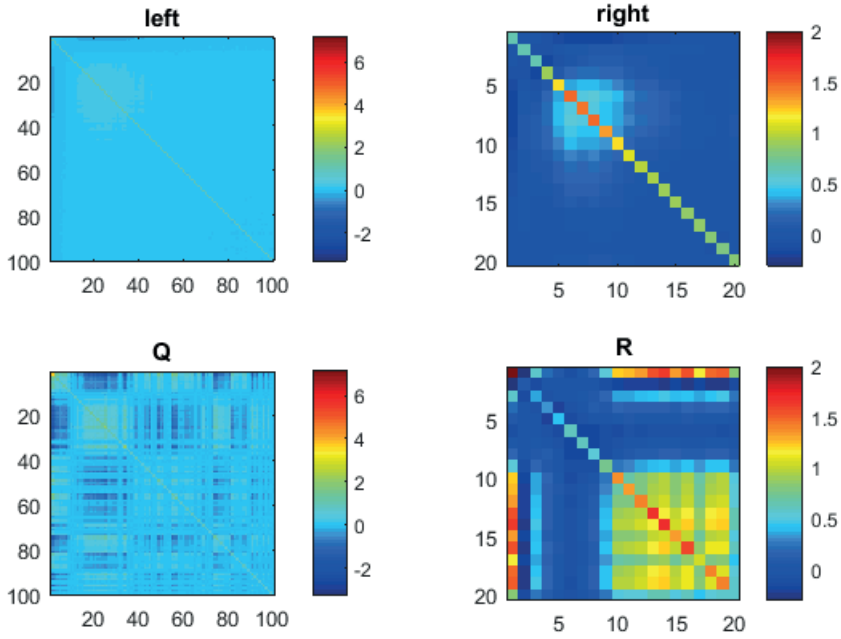

Figure 5.9: The resulting matrices $\mathbf{\Lambda}_{\text{left}}$ (first line left) and $\mathbf{\Lambda}_{\text{right}}$ (first line right) obtained by $\text{GLMQ}_{\text{l/r-MRL}}$ for an example run for the R-TRLF data set. In the second line the relevance matrices $\mathbf{Q}$ (left) and $\mathbf{R}$ (right) are pictured .

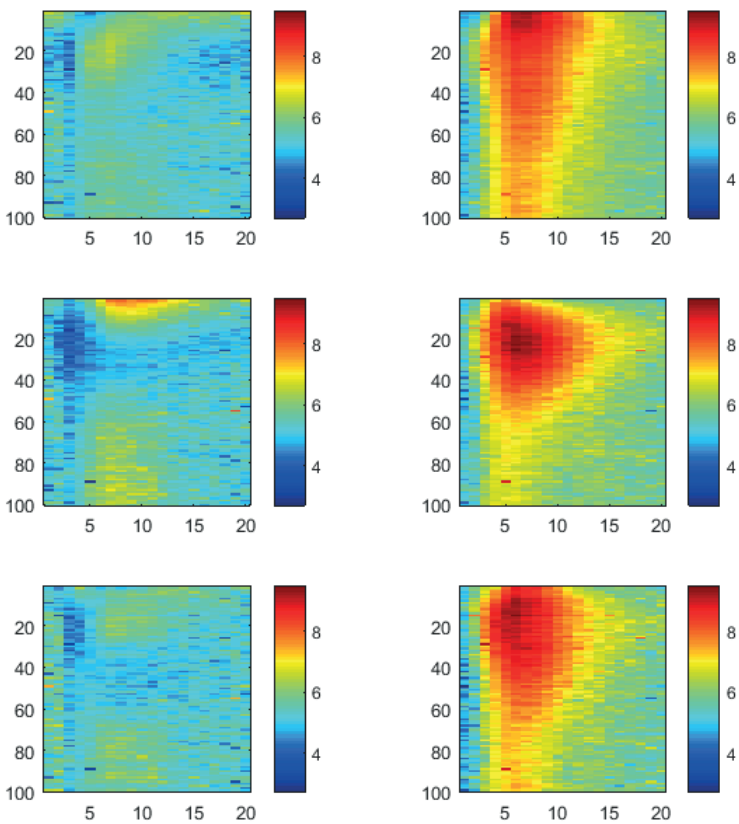

Figure 5.10: Prototypes for the two classes (excitation energy levels) obtained for an example run of the R-TRLF data set by using $\mathrm{GLMQ}_{\text{l-MRL}}$ (first line), $\mathrm{GLMQ}_{\text{r-MRL}}$ (second line) and $\mathrm{GLMQ}_{\text{QR}}$ (last line).

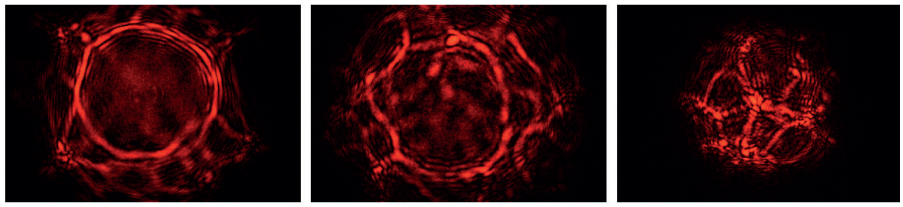

Figure 5.11: LDL-images of bacteria colonies of Salmonella serovars: Salmonella Brandenburg, Salmonella Enteritidis, and Salmonella Thyphimurium (from left to right).

Application in Salmonella Serovars Based on Laser-Diffuse-Light-Images

For a further exemplary application a data set of *Laser-Diffuse-Light-Images* (LDL-images) of Salmonella serovars colonies is considered. This data set consists of images from three salmonella types Salmonella Brandenburg (SB), Salmonella Enteritidis (SE), and Salmonella Thyphimurium (ST). The serovars have to be distinguished according the different structure of their colonies, see Figure 5.11. After preprocessing including centering, normalization, and calibration, the gray-scale-images have the size 128×128 pixel. In all, the data set, denoted as SALMONELLA, consists of 65 SB-samples, 48 SE-samples and 50 ST-samples. For this data also a vectorized version is generated, termed SALMONELLA-V. More details concerning this data can be found in [94].

Simulation Results of Salmonella Serovars The salmonella serovars images are investigated by means of GLMQ without and with several kinds of relevance learning. Only one prototype per class and a constant learning rate $\eta = 0.01$ is used, where 1000 learning epochs for each run are applied. The resulting accuracies and standard deviations are summarized in Table 5.4.Accordingly, the achieved performances have no significant difference, i. e. several kinds of relevance learning of GLMQ compared among themselves have no considerable effect on the classification performance. However, GLMQ with relevance learning achieves an essential improvement in comparison to the basis approach of GLMQ without relevance learning. This result is also reflected by visualizing the prototypes and relevances matrices of several approaches. Precisely, the resulting prototypes for $\mathrm{GLMQ}_{\text{l-MRL}}$, $\mathrm{GLMQ}_{\text{r-MRL}}$, and $\mathrm{GLMQ}_{\text{QR}}$ are shown in Figure 5.12. Obviously, the first prototype obtained by $\mathrm{GLMQ}_{\text{QR}}$ looks more similar to salmonella type SB in comparison to the other resulting prototypes using $\mathrm{GLMQ}_{\text{l/r-MRL}}$. The respective relevance matrices for these methods are depicted in

data/method	GLMQ	GLMQ_{HRL}	$\text{GLMQ}_{\text{r-MRL}}$	$\text{GLMQ}_{\text{l-MRL}}$	GLMQ_{QR}
accuracy	87.7	89.6	92.7	92.1	92.0
standard deviation	0.020	0.012	0.036	0.028	0.028

Table 5.4: Averaged test accuracies in % and standard deviations for the SALMONELLA data using different GLMQ variants without and with several kinds of relevance learning.

Figure 5.13. Further, also GLVQ, GRLVQ and GMLVQ are applied to the vectorized SALMONELLA-V data. Note that, there are extreme numerical instabilities for GMLVQ due to the huge classification correlation matrix $\mathbf{\Lambda} = \mathbf{\Omega}^\top \mathbf{\Omega} \in \mathbb{R}^{16384 \times 16384}$. In that case, another indicator for the pure model validity is the discrepancy of the test accuracy to the training performance of 86%, which indicates a very weak generalization ability of the model.

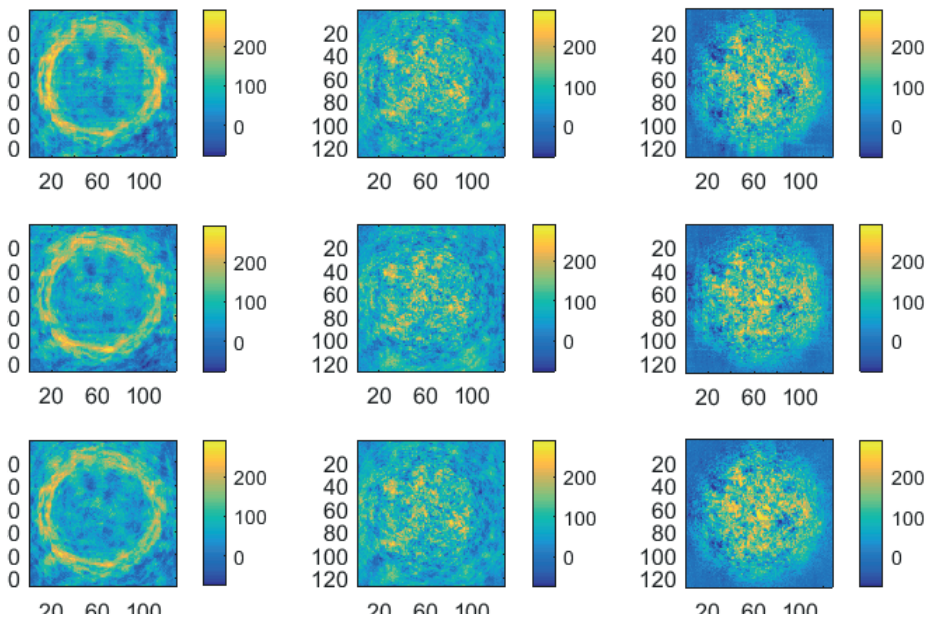

Figure 5.12: Prototypes (from left: Salmonella Brandenburg, Salmonella Enteritidis, and Salmonella Thyphimurium) obtained for an example run for the SALMONELLA data set by using $\text{GLMQ}_{\text{l-MRL}}$ (first line), $\text{GLMQ}_{\text{r-MRL}}$ (second line) and GLMQ_{QR} (last line).

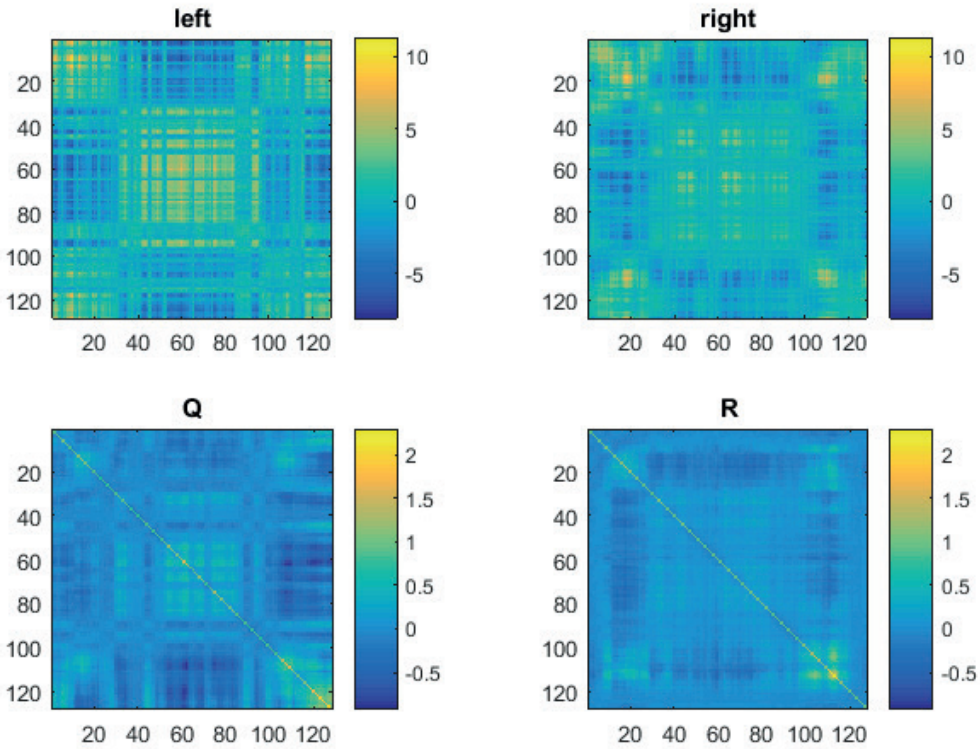

Figure 5.13: In the first line the matrices $\mathbf{\Lambda}_{\text{left}}$ and $\mathbf{\Lambda}_{\text{right}}$ obtained by $\text{GLMQ}_{\text{l/r-MRL}}$ for an example run for the SALMONELLA data set are pictured. The relevance matrices $\mathbf{Q}$ (left) and $\mathbf{R}$ (right) by means of GLMQ_{QR} are shown in the second line.

Chapter 6

Summary and Concluding Remarks

This thesis provides an unified and generalized scheme for Hebbian approaches in non-Euclidean spaces for unsupervised and supervised learning. Here the Hebbian learning methods are distinguished between vectorial and matrix approaches.

Hebbian learning methods for vectorial approaches Commonly, Hebbian like learning methods, such as Hebbian PCA learning, uses the Euclidean inner product for data processing. Thus, in this thesis Hebbian learning methods based on general inner products were defined and the focus was on Hebbian methods for unsupervised PCA and ICA. The key idea was the replacement of the Euclidean inner product by a more general SIP to equip Hebbian methods with a non-Euclidean metric. For that, in chapter 3 a framework for metrics induced by norms, which are generated by SIPs, was provided. These SIPs are the natural equivalent of the inner products for Banach spaces. A closer look was taken at known examples of Banach spaces, i. e. l_p-spaces and l_p-Sobolev-spaces with their SIPs were considered. Whereby for the l_p-Sobolev-space the respective SIP was proposed. Further, inner products and SIPs in general kernel spaces are also addressed, which allow the application of kernel metrics with underlying RKHS and RKBS.

The SIPs can be directly plugged into the above mentioned Hebbian learning approaches. The theoretical framework for non-Euclidean PCA and ICA by Hebbian learning was provided in chapter 4. Resulting methods are adaptive PCA by Hebbian

learning for general finite-dimensional Banach and Hilbert spaces as well as Kernel PCA by Hebbian Learning for RKHS and RKBS. By generalizing the original Hebbian PCA learning in this way, the gap between kernel-based learning and adequate data visualization is closed when kernel learning is performed with differentiable kernels that enable prototype learning with differentiable kernel metrics in the data space. This is also valid for other Non-Euclidean PCA variants, which are based on l_p-norms or Sobolev-norms.

Further, the idea of Hebbian learning for kernel PCA in non-Euclidean spaces was transferred to Hebbian ICA learning realizing an non-Euclidean ICA. The proposed learning rule assumes whitened data and it should be emphasized that pre-whitening has to be equipped with the respective kernel metric to ensure the stability of the ICA algorithm. Hence, Kernel PCA by Hebbian Learning in RKHS constitutes an appropriate preprocessing step for kernel ICA by Hebbian learning in RKHS. At the end of chapter 4 exemplary applications have shown that this non-Euclidean variant of Hebbian-like ICA is capable of extracting non-linearly mixed signals. In addition at the end of chapter 4 example applications and simulations for Non-Euclidean PCA were demonstrated and the results are compared with those of the Euclidean PCA. The exemplary applications for PCA and ICA show that, depending on the data and the type of application, Hebbian learning approaches based on a general inner product instead of the standard Euclidean inner product may be more suitable for data processing.

The LVQ algorithms, which are also Hebbian-like learning methods, are addressed in chapter 5 for non-Euclidean spaces. A non-Euclidean variant of GLVQ with l_p-norms requires the derivatives of them, which are at the origin ($x = 0$) not possible due to the absolute value function in the l_p-norms. Therefore, investigations of two smooth approximations of the maximum function referred to as α-softmax and α-quasimax function as well as the derivatives of them. It can be recommended that the α-quasimax function should be used to approximate the maximum function. Further, smooth approximations of the absolute value function, which are based on the maximum function, are also investigated and its derivatives. In general, it should be noted that the underlying approximation functions for l_p-norms applied in GLVQ, which includes the derivatives of the maximum function and the absolute value function, has to be consistent. The applications at the end of chapter 5 has shown that distances based on l_p-norms can be successfully applied by using the approximation functions as well as a consistent approach in GLVQ improves the classification performance in comparison to a non-consistent variant.

Hebbian learning methods for matrix approaches Another main part of this thesis were Hebbian approaches in non-Euclidean spaces of matrices. For that in the second part of chapter 3 the Banach spaces of matrices with their SIPs are investigated, where the focus was put on Schatten-p-norms. A SIP for this Banach space which generates the respective norm was developed. The **QR**-norm, which can be seen as an extended variant of the Schatten-p-norm, was examined more closely regarding the norm properties, i. e. properties introduced by G. Allen in [4] are adjusted. Further, an inner product for this Hilbert space, which generates the **QR**-norm, is proposed. Yet, the development of a SIP for the **QR**-norm with arbitrary $p \neq 2$ remains an open problem.

The resulting (semi)-inner products can be directly applied in the Hebbian PCA approach for matrices, whose theory and the proof of the required properties were provided in chapter 4. The resulting method is a Hebbian learning rule based on Schatten-p-norms for principal components in Banach spaces of matrices. In the last part of chapter 4 the different behavior of the matrix approach compared to the vectorial variant for an illustrative example was shown with the result, that in case of processing matrix data containing important spatial information the use of the matrix PCA leads to better performance.

Further, matrix norms can be also used as dissimilarity measure for supervised Hebbian like learning methods. In chapter 5 the LVQ approach was extended for the classification of matrix data using Schatten-p-norms and is called learning matrix quantization (LMQ). Compared to the vectorial counterpart, the use of matrix norms leads to a greater structural flexibility of relevance learning in GLMQ. The resulting methods based on several kinds of algebraic relevance weighting are GLMQ with Hadamard-Relevance-Learning ($\mathrm{GLMQ}_{\mathrm{HRL}}$) and left/right Multiplicative-Relevance-Learning ($\mathrm{GLMQ}_{\mathrm{l/r\text{-}MLR}}$)). Beside the Schatten-$p$-norm also the **QR**-norm was applied in GLMQ and referred to as QR-Relevance-Learning ($\mathrm{GLMQ}_{\mathrm{QR}}$). The Kronecker product between matrices and tensor composites using a relevance tensor instead of a relevance matrix form further kinds of algebraic compositions in GLMQ, which are denoted as Kronecker-Relevance-Learning (KRL) and Tensor-Relevance-Learning (TRL), respectively. One idea of applying these compositions was to get a counterpart of GMLVQ. The expected variability of KRL in GLMQ is degenerated to one parameter and, thus, $\mathrm{GLMQ}_{\mathrm{KRL}}$ is meaningless. A similar result was obtained by considering GLMQ with relevance tensors of fourth order, i. e. the respective relevance learning variability is the same as for $\mathrm{GLMQ}_{\mathrm{KRL}}$. In case of applying relevance tensors of third order the tensor operation result in a vector and, hence, this variant is also no counterpart of GMLVQ.

In summary, the advantage of LMQ is the greater flexibility of relevance learning compared to LVQ, but it is usually more complicated and generally difficult to interpret. This became evident by exemplary applications in the last part of chapter 5.

Future work

The properties of the relevance matrices of GLMQ should be investigated more closely in the application of various classification problems in order to obtain a better interpretation of them. Especially the interpretation of the relevance matrices **Q** and **R** when applying the **QR**-norm in GLMQ is still difficult, because the rows and column dependencies of a data point (matrix) are learned simultaneously. Further, the inner product of the **QR**-norm can also be applied in the matrix approach of Hebbian PCA learning. The benefit of the resulting method would be, that additional information of the data could be obtained due to the visualization with PCA based on the learned relevance matrices by GLMQ_{QR}. Further, the idea of the matrix approach of Hebbian PCA learning could be transferred to Hebbian ICA learning. Here, the independent components are defined in the Banach space of matrices. Obviously, the generalized scheme for Hebbian approaches in non-Euclidean spaces can also be used for realizing a non-Euclidean MCA.

Finally, the open problems mentioned above, i.e. the counterpart to GMLVQ in the matrix case and a SIP of the **QR**-norm, could be reinvestigated.

Further, for the presented matrix approaches there are potential application possibilities with some benefits. One interesting application could be LMQ with relevance learning for classification of fMRI data. In general, an fMRI data set consists of data matrices, where every matrix is assigned to one person [87]. Assuming that, the rows corresponds to voxels (measurement points in the brain) and the columns are the dimensions [87]. At first glance, there are two essential benefits. First, the data matrices can be processed without a previously vectorization, i. e. the spatial informations (voxel information as well as time dependencies) are preserved. The other benefit concerns the learned relevance matrices for a given classification task [2]. In detail, by applying GLMQ_{QR} the matrix **Q** will be responsible for the voxel and **R** for the time dependencies[1]. After learning, it could be obtained, how the voxels and, hence, brain regions are connected for a special classification task. Analogously, statements could also be made regarding the time dependencies of the fMRI sequences.

[1]Note that, the **QR**-norm has already been successfully applied for fMRI data by G. Allen [4].

Appendix A

Principle of Gradient Descent and Stochastic Gradient Descent Learning

Gradient descent (GD), along with its several variants, is probably the most widely used optimization technique in machine learning [127]. It is an iterative method for minimizing non-linear and non-convex optimization problems with the real valued objective (cost) function $E : \mathbb{R}^n \to \mathbb{R}$. Suppose $E(\Theta)$ is a cost function, where Θ represents the parameters. The GD starts with a randomly initialization of the parameters Θ and uses the gradient of the cost function with respect to Θ. The update rule of Θ at iteration $(t+1)$ is defined as

$$\Theta(t+1) = \Theta(t) - \eta \nabla E(\Theta(t)),$$

where η represents the learning rate, which is decreased with a decreasing rate $0 < \varsigma < 1$ by

$$\eta(\iota+1) = (1-\varsigma) \cdot \eta(\iota)$$

[99]. The iterative process is stopped after a fixed number of iterations or convergence is achieved, more on this later. This GD-principle is easy to implement and flexible for every kind of cost function. Only the differentiability of the cost function with respect to the parameters is assumed. Note, the GD principle is introduced for cost functions in the field of machine learning. Thus, the gradients in the GD-principle

are computed based on the whole training dataset. Thus, the gradient computation becomes expensive in case of huge datasets. An optimized method, the so called stochastic gradient descent (SGD), overcome this problem. SGD is a technique for minimizing a cost function based on the GD method, where the gradient at each step is estimated on the basis of a single randomly picked example (data point). In this thesis the SGD principle is considered for minimizing vector quantization cost functions

$$E\left(V, W, \boldsymbol{\varrho}\right) = \sum_{\mathbf{v} \in \mathbf{V}} L\left(\mathbf{v}, W, \boldsymbol{\varrho}\right) \tag{A.1}$$

with $L\left(\mathbf{v}, W, \boldsymbol{\varrho}\right)$ being the so-called (local) loss based on a single datapoint presentation $\mathbf{v} \in V$ with the prototypes W and further parameters $\boldsymbol{\varrho}$. According to this explanation $\frac{\partial_S E}{\partial \mathbf{w}} = \frac{\partial L}{\partial \mathbf{w}}$ is used as short hand notation for the stochastic gradient.[1] The SGD starts with a initialization of the parameters W and $\boldsymbol{\varrho}$ and a randomly chosen data point $\mathbf{v} \in V$ in each iteration step. All prototypes $\mathbf{w}_k \in W$ and further parameters $\varrho_l \in \boldsymbol{\varrho}$ are determined, which are responsible for $\mathbf{v}$. The update of the parameters is according to

$$\mathbf{w}_k \leftarrow \mathbf{w}_k - \eta_{\mathbf{w}}\left(\iota\right) \cdot \Delta \mathbf{w}_k \tag{A.2}$$

and

$$\varrho_l \leftarrow \varrho_l - \eta_{\boldsymbol{\varrho}}\left(\iota\right) \cdot \Delta \varrho_l \tag{A.3}$$

with the partial derivatives $\Delta \mathbf{w}_k = \frac{\partial L(\mathbf{v}, W, \boldsymbol{\varrho})}{\partial \mathbf{w}_k}$ and $\Delta \varrho_l = \frac{\partial L(\mathbf{v}, W, \boldsymbol{\varrho})}{\partial \varrho_l}$ $\forall k, l$, respectively. As above, the learning rates $\eta_{\mathbf{w}}$ in (A.2) and $\eta_{\boldsymbol{\varrho}}$ in (A.3) are decreased with a decreasing factor $\varsigma > 0$ by

$$\eta_{\mathbf{w}}\left(\iota + 1\right) = \left(1 - \varsigma\right) \cdot \eta_{\mathbf{w}}\left(\iota\right)$$

and

$$\eta_{\boldsymbol{\varrho}}\left(\iota + 1\right) = \left(1 - \varsigma\right) \cdot \eta_{\boldsymbol{\varrho}}\left(\iota\right),$$

respectively. This iterative process is repeated until convergence or manual stop. The convergence of the SGD to the global optima are secured by satisfying the conditions

$$\sum_{\iota} \eta^2\left(\iota\right) < \infty \tag{A.4}$$

[1] Note that, the optimization of the cost function (A.1) by the SGD principle includes the derivatives of the local costs $L\left(\mathbf{v}, W, \boldsymbol{\varrho}\right)$. Further, the continuous variant of (A.1) is $E\left(V, W, \boldsymbol{\varrho}\right) = \int_{\mathbf{v}} P\left(\mathbf{v}\right) L\left(\mathbf{v}, W, \boldsymbol{\varrho}\right) d\mathbf{v}$, where $P\left(\mathbf{v}\right)$ is the probability distribution of data vectors over V.

and

$$\sum_{\iota} \eta(\iota) = \infty \tag{A.5}$$

[77]. In particular, the learning rate should approach zero sufficiently fast (condition (A.4)) to damp out the noise effect as the iterate gets near the optima but should approach zero at a sufficiently (condition (A.5)) slow rate to avoid premature (false) convergence of the algorithm [127]. The SGD ends up at a local optimum and the handling of learning the parameters is demanding especially if more than one parameter is learned at the same time. However, the SGD is an established optimization technique in machine learning [128].

Appendix B

Proofs, Examples, Derivatives

B.1 Proof of the SIP of the Sobolev Space

According to Lemma (3.8) the SIP of $\mathcal{W}_{K,p}$ is given as

$$[f,g]_{\mathcal{W}_{K,p}} := \frac{1}{\|g\|^{p-2}_{\mathcal{W}_{K,p}}} \sum_{|\alpha|\leq K} \int D^\alpha f \, |D^\alpha g|^{p-1} \, \mathsf{sgn}\,(D^\alpha g)\, dt.$$

The SIP properties and uniqueness are proofed in the following.

Proof. **SIP properties:** According to Def. (3.1) the first two properties are obviously fulfilled. It remains to show Cauchy-Schwarz inequality. Consider

$$\begin{aligned}\left|[f,g]_{\mathcal{W}_{K,p}}\right| &= \left|\tfrac{1}{\|g\|^{p-2}_{\mathcal{W}_{K,p}}} \textstyle\sum_{|\alpha|\leq K} \int D^\alpha f \, |D^\alpha g|^{p-1} \, \mathsf{sgn}\,(D^\alpha g)\, dt\right| \\ &\leq \tfrac{1}{\|g\|^{p-2}_{\mathcal{W}_{K,p}}} \textstyle\sum_{|\alpha|\leq K} \int |D^\alpha f| \, |D^\alpha g|^{p-1} \, dt,\end{aligned}$$

where $1 < p < \infty$ and the triangle inequality was applied. By using the Hölder inequality for integrals it results

$$\frac{1}{\|g\|^{p-2}_{\mathcal{W}_{K,p}}} \sum_{|\alpha|\leq K} \int |D^\alpha f| \, |D^\alpha g|^{p-1} \, dt \leq \frac{1}{\|g\|^{p-2}_{\mathcal{W}_{K,p}}} \sum_{|\alpha|\leq K} \left(\int |D^\alpha f|^p \, dt\right)^{\frac{1}{p}} \left(\int |D^\alpha g|^{(p-1)q} \, dt\right)^{\frac{1}{q}} \tag{B.1}$$

for $\frac{1}{p} + \frac{1}{q} = 1$. Thus the right hand term in (B.1) can be rewritten with

$$q = \frac{p}{p-1} \tag{B.2}$$

as

$$\frac{1}{\|g\|^{p-2}_{\mathcal{W}_{K,p}}} \sum_{|\alpha|\leq K} \left(\int |D^\alpha f|^p \, dt \right)^{\frac{1}{p}} \left(\int |D^\alpha g|^{(p-1)q} \, dt \right)^{\frac{1}{q}} = \frac{1}{\|g\|^{p-2}_{\mathcal{W}_{K,p}}} \sum_{|\alpha|\leq K} \|D^\alpha f\|_p \cdot \|D^\alpha g\|^{p-1}_p \,. \tag{B.3}$$

A further majorization of (B.3) can be achieved by applying Hölder inequality for sums:

$$\frac{1}{\|g\|^{p-2}_{\mathcal{W}_{K,p}}} \sum_{|\alpha|\leq K} \|D^\alpha f\|_p \cdot \|D^\alpha g\|^{p-1}_p \leq \frac{1}{\|g\|^{p-2}_{\mathcal{W}_{K,p}}} \left(\sum_{|\alpha|\leq K} \|D^\alpha f\|^p_p \right)^{\frac{1}{p}} \cdot \left(\sum_{|\alpha|\leq K} \|D^\alpha g\|^{(p-1)q}_p \right)^{\frac{1}{q}} . \tag{B.4}$$

By recognizing the equivalence

$$\left(\sum_{|\alpha|\leq K} \|D^\alpha g\|^{(p-1)q}_p \right)^{\frac{1}{q}} = \|g\|_{\mathcal{W}_{K,p}} \tag{B.5}$$

and paying attention to the relation (B.2) again it can be conclude

$$\frac{1}{\|g\|^{p-2}_{\mathcal{W}_{K,p}}} \left(\sum_{|\alpha|\leq K} \|D^\alpha f\|^p_p \right)^{\frac{1}{p}} \cdot \left(\sum_{|\alpha|\leq K} \|D^\alpha g\|^{(p-1)q}_p \right)^{\frac{1}{q}} = \|f\|_{\mathcal{W}_{K,p}} \cdot \|g\|_{\mathcal{W}_{K,p}} \,. \tag{B.6}$$

which is the desired Cauchy-Schwarz inequality for the SIP of $\mathcal{W}_{K,p}$.

Uniqueness: The Sobolev space $\mathcal{W}_{K,p}$ can be seen as the Cartesian product

$$\mathcal{W}_{K,p} = \mathcal{L}^{(0)}_p \otimes \mathcal{L}^{(1)}_p \otimes \ldots \otimes \mathcal{L}^{(K)}_p$$

of $\mathcal{L}_p$-spaces $\mathcal{L}^{(k)}_p$ where k denotes the order of the derivative. Thus, there are the sum of uniformly convex spaces, which is uniformly convex itself. Then, the Remark (3.3) ensures the uniqueness. □

B.2 Example for the L_p^{TS} -Measure violating the triangle inequality

Let $\mathbf{x} = (0, 10, 1, 10, 1, 0)^\top$, $\mathbf{y} = (0, 10, -1, 10, -1, 0)^\top$ and it yields $\mathbf{x} + \mathbf{y} = (0, 20, 0, 20, 0, 0)^\top$. For $p = 2$ and $\tau = 1$ it results $L_p^{TS}(\mathbf{x}, \tau) \approx 14.21$, $L_p^{TS}(\mathbf{y}, \tau) \approx 13.19$ and $L_p^{TS}(\mathbf{x} + \mathbf{y}, \tau) \approx 28.28$. Therefore, $L_p^{TS}(\mathbf{x} + \mathbf{y}, \tau) > L_p^{TS}(\mathbf{x}, \tau) + L_p^{TS}(\mathbf{y}, \tau)$ is obtained, violating the triangle inequality.

B.3 Proof of SIP for the Schatten-p-norm:

According to proposition 3.15 on page 52 the SIP of the Schatten-p-norm can be state as follows:

$$[\mathbf{A}, \mathbf{B}]_{\mathcal{S}_p} = \frac{1}{\left(\|\mathbf{B}\|_{\mathcal{S}_p}\right)^{p-2}} \operatorname{trace}\left(\mathbf{A} \cdot \mathbf{B}^* \left(|\mathbf{B}|\right)^{p-2}\right)$$

Proof. The first proof was given in [81]. Here, we show the validity of the Cauchy-Schwarz-inequality in a more elegant way.

We start with a singular value decomposition $\mathbf{B} = \mathbf{U}\boldsymbol{\Sigma}_\mathbf{B}\mathbf{V}^*$ of $\mathbf{B}$ with unitary matrices $\mathbf{U}$ and $\mathbf{V}$. The diagonal matrix $\boldsymbol{\Sigma}_\mathbf{B}$ contains the singular values $\sigma_i(\mathbf{B})$ as diagonal elements. We consider the matrix $\mathbf{M} = \mathbf{B}^* |\mathbf{B}|^{p-2}$ and get

$$\begin{aligned}
\mathbf{M} &= \mathbf{B}^* \left(\sqrt{\mathbf{B}\mathbf{B}^*}\right)^{p-2} \\
&= \mathbf{V}\boldsymbol{\Sigma}_\mathbf{B}\mathbf{U}^* \left(\sqrt{\mathbf{U}\boldsymbol{\Sigma}_\mathbf{B}\mathbf{V}^*\mathbf{V}\boldsymbol{\Sigma}_\mathbf{B}\mathbf{U}^*}\right)^{p-2} \\
&= \mathbf{V}\boldsymbol{\Sigma}_\mathbf{B}\mathbf{U}^* \left(\sqrt{\mathbf{U}\left(\boldsymbol{\Sigma}_\mathbf{B}\right)^2\mathbf{U}^*}\right)^{p-2} \\
&= \mathbf{V}\boldsymbol{\Sigma}_\mathbf{B}\mathbf{U}^* \left(\mathbf{U}\boldsymbol{\Sigma}_\mathbf{B}\mathbf{U}^*\right)^{p-2} \\
&= \mathbf{V}\boldsymbol{\Sigma}_\mathbf{B}\mathbf{U}^*\mathbf{U}\left(\boldsymbol{\Sigma}_\mathbf{B}\right)^{p-2}\mathbf{U}^* \\
&= \mathbf{V}\left(\boldsymbol{\Sigma}_\mathbf{B}\right)^{p-1}\mathbf{U}^*
\end{aligned} \tag{B.7}$$

i.e. the singular values $\sigma_i(\mathbf{M})$ of $\mathbf{M}$ are obtained as $\sigma_i(\mathbf{M}) = (\sigma_i(\mathbf{B}))^{p-1}$ and, therefore,

$$\sigma_i(\mathbf{M}) = \sigma_i(\mathbf{B}) \cdot |\sigma_i(\mathbf{B})|^{p-2}$$

is valid. The next step uses the Neumann inequality

$$|\mathrm{tr}\,(\mathbf{AM})| \leq \sum_{k=1}^{\mu} \sigma_k\,(\mathbf{A}) \cdot \sigma_k\,(\mathbf{M}) \tag{B.8}$$

for matrices $\mathbf{A}, \mathbf{M} \in \mathbb{C}^{m\times n}$ and $\mu = \min\{m, n\}$, which yields

$$|\mathrm{tr}\,(\mathbf{AM})| \leq \sum_{k=1}^{\mu} \sigma_k\,(\mathbf{A}) \cdot \sigma_k\,(\mathbf{B}) \cdot |\sigma_k\,(\mathbf{B})|^{p-2} \tag{B.9}$$

in our case. We collect the singular values $\sigma_k\,(\mathbf{A})$ and $(\sigma_k\,(\mathbf{B}))$ into the vectors $(\boldsymbol{\sigma}\,(\mathbf{A}))$ and $(\boldsymbol{\sigma}\,(\mathbf{M}))$, respectively, and apply to them the SIP

$$[\mathbf{x}, \mathbf{y}]_{l_p} = \frac{1}{\left(\|\mathbf{y}\|_{l_p}\right)^{p-2}} \sum_{i=1}^{\mu} x_i \cdot \bar{y}_i \cdot |y_i|^{p-2}$$

for l_p-spaces corresponding to the Minkowski norm (3.5). Thus we have

$$[\boldsymbol{\sigma}\,(\mathbf{A}), \boldsymbol{\sigma}\,(\mathbf{B})]_{\mathcal{S}_p} = \frac{1}{\left(\|\boldsymbol{\sigma}\,(\mathbf{B})\|_{l_p}\right)^{p-2}} \sum_{k=1}^{\mu} \sigma_k\,(\mathbf{A}) \cdot \sigma_k\,(\mathbf{B}) \cdot |(\sigma_k\,(\mathbf{B}))|^{p-2} \tag{B.10}$$

satisfying the Cauchy-Schwarz-inequality

$$\left|[\boldsymbol{\sigma}\,(\mathbf{A}), \boldsymbol{\sigma}\,(\mathbf{B})]_{\mathcal{S}_p}\right|^2 \leq \|\boldsymbol{\sigma}\,(\mathbf{A})\|_{l_p} \cdot \|\boldsymbol{\sigma}\,(\mathbf{B})\|_{l_p} \tag{B.11}$$

because of the SIP-properties. Now we can collect all pieces together and obtain

$$\begin{aligned}
\left|[\mathbf{A},\mathbf{B}]_{\mathcal{S}_p}\right|^2 &= \left|\frac{1}{\left(\|\mathbf{B}\|_{\mathcal{S}_p}\right)^{p-2}}\mathrm{tr}\left(\mathbf{A}\mathbf{B}^*\left|\mathbf{B}\right|^{p-2}\right)\right|^2 \\
&= \left(\frac{1}{\left(\|\mathbf{B}\|_{\mathcal{S}_p}\right)^{p-2}}\left|\mathrm{tr}\left(\mathbf{A}\mathbf{B}^*\left|\mathbf{B}\right|^{p-2}\right)\right|\right)^2 \\
&\overset{(B.7)}{=} \left(\frac{1}{\left(\|\boldsymbol{\sigma}\left(\mathbf{B}\right)\|_{l_p}\right)^{p-2}}\left|\mathrm{tr}\left(\mathbf{A}\mathbf{M}\right)\right|\right)^2 \\
&\overset{(B.9)}{\leq} \left(\frac{1}{\left(\|\boldsymbol{\sigma}\left(\mathbf{B}\right)\|_{l_p}\right)^{p-2}}\sum_{k=1}^{\mu}\sigma_k\left(\mathbf{A}\right)\cdot\sigma_k\left(\mathbf{B}\right)\cdot\left|\sigma_k\left(\mathbf{B}\right)\right|^{p-2}\right)^2 \\
&\overset{(B.10)}{=} \left|\left[\boldsymbol{\sigma}\left(\mathbf{A}\right),\boldsymbol{\sigma}\left(\mathbf{B}\right)\right]_{\mathcal{S}_p}\right|^2 \\
&\overset{(B.11)}{\leq} \left(\|\boldsymbol{\sigma}\left(\mathbf{A}\right)\|_{l_p}\right)^2\cdot\left(\|\boldsymbol{\sigma}\left(\mathbf{B}\right)\|_{l_p}\right)^2 \\
&= \left(\|\mathbf{A}\|_{\mathcal{S}_p}\right)^2\cdot\left(\|\mathbf{B}\|_{\mathcal{S}_p}\right)^2 \\
&= \left[\mathbf{A},\mathbf{A}\right]_{\mathcal{S}_p}\left[\mathbf{B},\mathbf{B}\right]_{\mathcal{S}_p}
\end{aligned}$$

completing the proof. □

B.4 Proof of Lemma 3.14

Lemma. *The Schatten-p-norm is unitarily invariant, i.e* $\|\mathbf{A}\cdot\mathbf{U}\|_{\mathcal{S}_p} = \|\mathbf{A}\|_{\mathcal{S}_p}$ *with* $\mathbf{U}$ *being an unitary matrix.*

Proof. Consider $\|\mathbf{A}\|_{\mathcal{S}_p} = \sqrt[p]{\operatorname{tr}\left((\mathbf{A}\mathbf{A}^*)^p\right)}$ and get

$$\begin{aligned}
\|\mathbf{A}\cdot\mathbf{U}\|_{\mathcal{S}_p} &= \sqrt[p]{\operatorname{tr}\left(\left((\mathbf{A}\cdot\mathbf{U})\,(\mathbf{A}\cdot\mathbf{U})^*\right)^{\frac{p}{2}}\right)} \\
&= \sqrt[p]{\operatorname{tr}\left((\mathbf{A}\cdot\mathbf{U}\cdot\mathbf{U}^*\cdot\mathbf{A}^*)^{\frac{p}{2}}\right)} \\
&= \sqrt[p]{\operatorname{tr}\left((\mathbf{A}\cdot\mathbf{I}\cdot\mathbf{A}^*)^{\frac{p}{2}}\right)} \\
&= \|\mathbf{A}\|_{\mathcal{S}_p}
\end{aligned}$$

as requested. Similarly, the relation $\|\mathbf{U}\cdot\mathbf{A}\|_{\mathcal{S}_p} = \|\mathbf{A}\|_{\mathcal{S}_p}$ can be proofed using the equality $\sqrt[p]{\operatorname{tr}\left((\mathbf{A}\mathbf{A}^*)^{\frac{p}{2}}\right)} = \sqrt[p]{\operatorname{tr}\left((\mathbf{A}^*\mathbf{A})^{\frac{p}{2}}\right)}$. □

B.5 Proof of Lemma 3.17

Lemma. *The norm $\|\mathbf{A}\|_{\mathbf{Q},\mathbf{R}}$ from (3.42) constitutes a vector norm for matrices for positive definite symmetric matrices $\mathbf{Q}$ and $\mathbf{R}$.*

Proof. For $\mathbf{Q} = \mathbf{R} = \mathbf{I}$ we get

$$\|\mathbf{A}\|_{\mathbf{I},\mathbf{I}} = \sqrt{\operatorname{tr}\left(\mathbf{A}\mathbf{A}^*\right)} = s_2\left(\mathbf{A}\right)$$

being the Frobenius norm. In the next step we suppose regular matrices $\tilde{\mathbf{Q}}$ and $\tilde{\mathbf{R}}$, we can conclude that

$$s_2\left(\tilde{\mathbf{Q}}\mathbf{A}\tilde{\mathbf{R}}\right) = \sqrt{\operatorname{tr}\left(\left(\tilde{\mathbf{Q}}\mathbf{A}\tilde{\mathbf{R}}\right)\left(\tilde{\mathbf{Q}}\mathbf{A}\tilde{\mathbf{R}}\right)^*\right)}$$

is a vector norm for matrices according to [50, p. 371, Example 1]. Now we have

$$\begin{aligned}
\sqrt{\operatorname{tr}\left(\left(\tilde{\mathbf{Q}}\mathbf{A}\tilde{\mathbf{R}}\right)\left(\tilde{\mathbf{Q}}\mathbf{A}\tilde{\mathbf{R}}\right)^*\right)} &= \sqrt{\operatorname{tr}\left(\tilde{\mathbf{Q}}\mathbf{A}\tilde{\mathbf{R}}\tilde{\mathbf{R}}^*\mathbf{A}^*\tilde{\mathbf{Q}}^*\right)} \\
&= \sqrt{\operatorname{tr}\left(\mathbf{A}\tilde{\mathbf{R}}\tilde{\mathbf{R}}^*\mathbf{A}^*\tilde{\mathbf{Q}}^*\tilde{\mathbf{Q}}\right)} \\
&= \sqrt{\operatorname{tr}\left(\mathbf{A}\mathbf{R}\mathbf{A}^*\tilde{\mathbf{Q}}^*\tilde{\mathbf{Q}}\right)} \\
&= \sqrt{\operatorname{tr}\left(\mathbf{Q}\mathbf{A}\mathbf{R}\mathbf{A}^*\right)}
\end{aligned}$$

with

$$\mathbf{Q} = \tilde{\mathbf{Q}}^* \tilde{\mathbf{Q}} \text{ and } \mathbf{R} = \tilde{\mathbf{R}} \tilde{\mathbf{R}}^* \tag{B.12}$$

as positive definite matrices and using the cyclic property of the trace operator. Hence, $\|\mathbf{A}\|_{\mathbf{Q},\mathbf{R}} = s_2\left(\tilde{\mathbf{Q}}\mathbf{A}\tilde{\mathbf{R}}\right)$ is valid and, therefore, $\|\mathbf{A}\|_{\mathbf{Q},\mathbf{R}}$ is a norm. □

B.6 Proof of Lemma 3.19

Lemma. *The norm* $\|\mathbf{A}\|_{\mathbf{Q},\mathbf{R}}$ *is unitarily invariant iff either* $\mathbf{R} = \mathbf{I}$ *or* $\mathbf{Q} = \mathbf{I}$ *is valid.*

Proof. For $\mathbf{R} = \mathbf{I}$ we have

$$\begin{aligned} \|\mathbf{A} \cdot \mathbf{U}\|_{\mathbf{Q},\mathbf{I}} &= \sqrt{\operatorname{tr}\left(\mathbf{Q}\left(\mathbf{A} \cdot \mathbf{U}\right)\mathbf{I}\left(\mathbf{A} \cdot \mathbf{U}\right)^*\right)} \\ &= \sqrt{\operatorname{tr}\left(\mathbf{Q}\mathbf{A} \cdot \mathbf{U} \cdot \mathbf{U}^* \cdot \mathbf{A}^*\right)} \\ &= \sqrt{\operatorname{tr}\left(\mathbf{Q}\mathbf{A} \cdot \mathbf{A}^*\right)} \\ &= \|\mathbf{A}\|_{\mathbf{Q},\mathbf{I}} \end{aligned}$$

and for $\mathbf{Q} = \mathbf{I}$ we calculate

$$\begin{aligned} \|\mathbf{U} \cdot \mathbf{A}\|_{\mathbf{I},\mathbf{R}} &= \sqrt{\operatorname{tr}\left(\left(\mathbf{U} \cdot \mathbf{A}\right)\mathbf{R}\left(\mathbf{U} \cdot \mathbf{A}\right)^*\right)} \\ &= \sqrt{\operatorname{tr}\left(\mathbf{U} \cdot \mathbf{A} \cdot \mathbf{R} \cdot \mathbf{A}^* \cdot \mathbf{U}^*\right)} \\ &= \sqrt{\operatorname{tr}\left(\mathbf{U}^* \cdot \mathbf{U} \cdot \mathbf{A} \cdot \mathbf{R} \cdot \mathbf{A}^*\right)} \\ &= \sqrt{\operatorname{tr}\left(\mathbf{A} \cdot \mathbf{R} \cdot \mathbf{A}^*\right)} \\ &= \|\mathbf{A}\|_{\mathbf{I},\mathbf{R}} \end{aligned}$$

which completes the proof. □

B.7 Proof of Lemma 3.20

Lemma. *The norm* $\|\mathbf{A}\|_{\mathbf{Q},\mathbf{R}}$ *is generated by an inner product defined by*

$$\langle \mathbf{A}, \mathbf{B} \rangle_{\mathbf{Q},\mathbf{R}} = \operatorname{tr}\left(\mathbf{Q}\mathbf{A}\mathbf{R}\mathbf{B}^*\right) \tag{B.13}$$

for positive definite symmetric matrices $\mathbf{Q}$ *and* $\mathbf{R}$*, i.e. also the Hermitian symmetries* $\mathbf{Q} = \mathbf{Q}^*$ *and* $\mathbf{R} = \mathbf{R}^*$ *are valid.*

Proof. For the proof of the lemma we have to show the inner product properties for $\langle \mathbf{A}, \mathbf{B} \rangle_{\mathbf{Q},\mathbf{R}}$:

1. **positive definiteness**

$$\begin{aligned} \langle \mathbf{A}, \mathbf{A} \rangle_{\mathbf{Q},\mathbf{R}} &= \mathrm{tr}\left(\mathbf{QARA}^*\right) \\ &= \|\mathbf{A}\|^2_{\mathbf{Q},\mathbf{R}} \\ &\geq 0 \end{aligned}$$

and the zero value is obviously obtained only for $\mathbf{A} = \mathbf{0}$.

2. **Hermitian symmetry**

To show the property we use the Hermitian symmetry of the trace operator, i.e $\mathrm{tr}\left(\mathbf{CD}^*\right) = \overline{\mathrm{tr}\left(\left(\mathbf{CD}^*\right)^*\right)}$ with the settings $\mathbf{C} = \mathbf{QA}$ and $\mathbf{D}^* = \mathbf{RB}^*$. We derive

$$\begin{aligned} \langle \mathbf{A}, \mathbf{B} \rangle_{\mathbf{Q},\mathbf{R}} &= \mathrm{tr}\left(\mathbf{QARB}^*\right) \\ &= \mathrm{tr}\left(\mathbf{CD}^*\right) \\ &= \overline{\mathrm{tr}\left(\mathbf{DC}^*\right)} \\ &= \overline{\mathrm{tr}\left(\mathbf{BR}^*\mathbf{A}^*\mathbf{Q}^*\right)} \\ &= \overline{\mathrm{tr}\left(\mathbf{Q}^*\mathbf{BR}^*\mathbf{A}^*\right)} \\ &= \overline{\langle \mathbf{B}, \mathbf{A} \rangle_{\mathbf{Q}^*,\mathbf{R}^*}} \\ &= \overline{\langle \mathbf{B}, \mathbf{A} \rangle_{\mathbf{Q},\mathbf{R}}} \end{aligned}$$

where the Hermitian symmetry of $\mathbf{Q}$ and $\mathbf{R}$ was used in the last line.

3. **Linearity for the first argument**

$$\begin{aligned} \langle \lambda\mathbf{A} + \mu\mathbf{C}, \mathbf{B} \rangle_{\mathbf{Q},\mathbf{R}} &= \mathrm{tr}\left(\mathbf{Q}\left(\lambda\mathbf{A} + \mu\mathbf{C}\right)\mathbf{RB}^*\right) \\ &= \mathrm{tr}\left(\lambda\mathbf{QARB}^* + \mu\mathbf{QCRB}^*\right) \\ &= \lambda\mathrm{tr}\left(\mathbf{QARB}^*\right) + \mu\mathrm{tr}\left(\mathbf{QCRB}^*\right) \end{aligned}$$

using the linearity of the trace operator.

4. **Homogeneity for the second argument**

$$\begin{aligned}\langle \mathbf{A}, \lambda\mathbf{B}\rangle_{\mathbf{Q},\mathbf{R}} &= \operatorname{tr}\left(\mathbf{QAR}\left(\lambda\mathbf{B}\right)^*\right)\\ &= \operatorname{tr}\left(\mathbf{QAR}\overline{\lambda}\mathbf{B}^*\right)\\ &= \overline{\lambda}\operatorname{tr}\left(\mathbf{QARB}^*\right)\end{aligned}$$

Further, $\|\mathbf{A}\|_{\mathbf{Q},\mathbf{R}} = \sqrt{\langle \mathbf{A}, \mathbf{A}\rangle_{\mathbf{Q},\mathbf{R}}}$ holds because of the positive definiteness of the inner product, i.e. $\langle \mathbf{A}, \mathbf{B}\rangle_{\mathbf{Q},\mathbf{R}}$ generates the norm $\|\mathbf{A}\|_{\mathbf{Q},\mathbf{R}}$. □

B.8 Formal Derivatives of l_p-norms for $p = \infty$

The formal element wise derivative of the prototype update becomes

$$\begin{aligned}\frac{\partial d_\infty\left(\mathbf{v},\mathbf{w}\right)}{\partial w_k} &= \frac{\partial \max\left(\left|v_k - w_k\right|\right)}{\partial\left(v_k - w_k\right)} \cdot \frac{\partial v_k - w_k}{w_k}\\ &= -\frac{\partial \max\left(\left|v_k - w_k\right|\right)}{\partial\left(v_k - w_k\right)}\end{aligned}$$

where the derivatives of weighted variants $d^{\boldsymbol{\lambda}}_\infty\left(\mathbf{v},\mathbf{w}\right)$ and $d^{\boldsymbol{\Omega}}_\infty\left(\mathbf{v},\mathbf{w}\right)$ reads as

$$\frac{\partial d^{\boldsymbol{\lambda}}_\infty\left(\mathbf{v},\mathbf{w}\right)}{\partial w_k} = -\frac{\partial \max\left(\lambda_k\left|v_k - w_k\right|\right)}{\partial\left(\lambda_k\left|v_k - w_k\right|\right)} \cdot \frac{\partial\left(\lambda_k\left|v_k - w_k\right|\right)}{\partial\left(v_k - w_k\right)}$$

and

$$\frac{\partial d^{\boldsymbol{\Omega}}_\infty\left(\mathbf{v},\mathbf{w}\right)}{\partial w_k} = -\Omega^\top_{i,k}\frac{\partial \max\left(\left|\boldsymbol{\Omega}\left(\mathbf{v} - \mathbf{w}\right)\right|\right)}{\partial \sum_{i=1}^n \Omega_{i,k}\left(v_k - w_k\right)} \cdot \frac{\partial\left(\left|\boldsymbol{\Omega}\left(\mathbf{v} - \mathbf{w}\right)\right|\right)}{\partial \sum_{i=1}^n \Omega_{i,k}\left(v_k - w_k\right)},$$

respectively. Further, element wise derivatives of the relevance update with respect to $\boldsymbol{\lambda}$ and $\boldsymbol{\Omega}$ yields

$$\begin{aligned}\frac{\partial d^{\boldsymbol{\lambda}}_\infty\left(\mathbf{v},\mathbf{w}\right)}{\partial \lambda_k} &= \frac{\partial \max\left(\boldsymbol{\lambda} \circ \left|\mathbf{v} - \mathbf{w}\right|\right)}{\partial\left(\lambda_k\left|v_k - w_k\right|\right)} \cdot \frac{\partial \lambda_k\left|v_k - w_k\right|}{\partial \lambda_k}\\ &= \frac{\partial \max\left(\boldsymbol{\lambda} \circ \left|\mathbf{v} - \mathbf{w}\right|\right)}{\partial\left(\lambda_k\left|v_k - w_k\right|\right)} \cdot \left|v_k - w_k\right|\end{aligned}$$

and

$$\begin{aligned}\frac{\partial d_{\infty}^{\mathbf{\Omega}}(\mathbf{v},\mathbf{w})}{\partial \Omega_{k,l}} &= \frac{\partial \max\left(\left|\mathbf{\Omega}\left(\mathbf{v}-\mathbf{w}\right)\right|\right)}{\partial\left|\mathbf{\Omega}\left(\mathbf{v}-\mathbf{w}\right)\right|} \cdot \frac{\partial\left(\left|\mathbf{\Omega}\left(\mathbf{v}-\mathbf{w}\right)\right|\right)}{\partial \sum_{i=1}^{n} \Omega_{i,k}\left(v_i-w_i\right)} \cdot \frac{\partial \sum_{i=1}^{n} \Omega_{i,k}\left(v_i-w_i\right)}{\Omega_{k,l}} \\ &= \frac{\partial \max\left(\left|\mathbf{\Omega}\left(\mathbf{v}-\mathbf{w}\right)\right|\right)}{\partial\left|\mathbf{\Omega}\left(\mathbf{v}-\mathbf{w}\right)\right|} \cdot \frac{\partial\left(\left|\mathbf{\Omega}\left(\mathbf{v}-\mathbf{w}\right)\right|\right)}{\partial \sum_{i=1}^{n} \Omega_{i,k}\left(v_i-w_i\right)} \cdot \left|v_k-w_k\right|\end{aligned}$$

respectively.

norms	Formula	(Semi-) inner product
l_p-norm	$\|\mathbf{x}\|_{l_p} = \sqrt[p]{\sum_{i=1}^n \|x_i\|^p}$	$[\mathbf{x},\mathbf{y}]_{l_p} = \frac{1}{\left(\|\mathbf{y}\|_{l_p}\right)^{p-2}} \sum_{i=1}^n x_i \|y_i\|^{p-1} \operatorname{sign}(y_i)$
$\hat{l_p}$–norm	$\|\mathbf{x}\|_{\hat{l_p}} = \sqrt[p]{\sum_{i=1}^n \|x_i\|^p}$	$[\mathbf{x},\mathbf{y}]_{\hat{l_p}} = \frac{1}{\left(\|\mathbf{y}\|_{l_p}\right)^{p-2}} \sum_{i=1}^n x_i \bar{y_i} \|y_i\|^{p-2}$
l_1-norm	$\|\mathbf{x}\|_{l_1} = \sum_{i=1}^n \|x_i\|$	$[\mathbf{x},\mathbf{y}]_{l_1} = \|\mathbf{y}\|_{l_1} \sum_{i=1}^n x_i \cdot \operatorname{sign}(y_i)$
l_2-norm	$\|\mathbf{x}\|_{l_2} = \sqrt{\sum_{i=1}^n x_i^2}$	$[\mathbf{x},\mathbf{y}]_{l_2} = \sum_{i=1}^n x_i y_i$
l_∞-norm	$\|\mathbf{x}\|_{l_\infty} = \sup_i \|x_i\|$	not known
Sobolev-norm	$\|f\|_{\mathcal{W}_{K,p}} = \left(\sum_{\|\alpha\|\leq K} \int \|D^\alpha f\|^p \, dt\right)^{\frac{1}{p}}$	$[f,g]_{\mathcal{W}_{K,p}} := \frac{1}{\|g\|^{p-2}_{\mathcal{W}_{K,p}}} \sum_{\|\alpha\|\leq K} \int D^\alpha f \, \|D^\alpha g\|^{p-1} \operatorname{sign}(D^\alpha g) \, dt.$
Schatten-p-norm	$\|\mathbf{A}\|_{\mathcal{S}_p} = \sqrt[p]{\operatorname{tr}\left(\|\mathbf{A}\|^p\right)}$	$[\mathbf{A},\mathbf{B}]_{\mathcal{S}_p} = \frac{1}{\left(\|\mathbf{B}\|_{\mathcal{S}_p}\right)^{p-2}} \operatorname{tr}\left(\mathbf{A}\cdot\mathbf{B}^* \left(\|\mathbf{B}\|_m\right)^{p-2}\right)$
QR-norm	$\|\mathbf{A}\|_{\mathcal{S}_2,\mathbf{Q},\mathbf{R}} = \sqrt{\operatorname{tr}\left(\mathbf{QARA}^*\right)}$	$[\mathbf{A},\mathbf{B}]_{\mathcal{S}_2\mathbf{Q},\mathbf{R}} = \operatorname{tr}\left(\mathbf{QARB}^*\right)$

Table B.1: Summary of vector- and matrix norms and their (semi-) inner products considered in this thesis.

My Publications

Publications related to this thesis

1. *M. Lange and M. Biehl and T. Villmann,* "Non-Euclidean Principal Component Analysis by Hebbian Learning", *Neurocomputing 147 (2015), pp. 107-119[83]*

2. *M. Biehl, M. Kästner, M. Lange and T. Villmann,* "Non-Euclidean Principal Component Analysis and Oja's Learning Rule - Theoretical Aspects", *in P.A. Estevez and J.C. Principe and P. Zegers, ed., Advances in Self-Organizing Maps: 9th International Workshop WSOM 2012 Santiage de Chile vol. 198, (Berlin: Springer, 2013), pp. 23-34.[14]*

3. *T. Villmann and M. Lange*, "A comment on the functional L_p^{TS}-Measure Regarding the norm properties", *TechReport, 2015.[140]*

4. *K. Domaschke, M. Kaden, M. Lange, T. Villmann,* "Learning Matrix Quantization and Variants of Relevance Learning", *in M. Verleysen, ed., Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2015), pp. 13-18, Louvain-La-Neuve, Belgium (2015). [32]*

5. *A. Bohnsack, K. Domaschke, M. Kaden, M. Lange and T. Villmann,* "Learning Matrix Quantization and Relevance Learning Based on Schatten-p-norms", *Neurocomputing 192 (2016), pp. 104-114.[19]*

6. *A. Bohnsack, K. Domaschke, M. Kaden, M. Lange and T. Villmann,* "Mathematical Characterization of Sophisticated Variants for Relevance Learning in Learning Matrix Quantization Based on Schatten-p-norms", *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science) 1 (2015), pp. 403-414.[18]*

7. *M. Lange, M. Biehl, T. Villmann,* "Non-Euclidean Independent Component Analysis and Oja's Learning", *in M. Verleysen, ed., Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2013) (Louvain-La-Neuve, Belgium: i6doc.com, 2013), pp. 125-130.[80]*

8. *M. Lange, D. Nebel and T. Villmann, "*Non-Euclidean Principal Component Analysis for Matrices by Hebbian Learning*", in L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh and J.M. Zurada, ed., Artificial Intelligence and Soft Computing - Proc. the International Conference ICAISC vol. 8467, (Zakopane: Springer, 2014), pp. 77-88.[81]*

9. *M. Lange, T. Villmann*, "Derivatives of l_p-norms and their Approximations", *Machine Learning Reports* 7, *MLR-04-2013 (2013), pp. 43-59.[79]*

10. *M. Lange, D. Zühlke, O. Holz, T. Villmann*, "Applications of l_p-norms and their Smooth Approximations for Gradient Based Learning Vector Quantization", *in M. Verleysen, ed., Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2014),pp. 271-276. [82]*

11. *M. Kaden, M. Lange, D. Nebel, M. Riedel and T. Geweniger and T. Villmann, "*Aspects in Classification Learning - Review of Recent Developments in Learning Vector Quantization*", Foundations of Computing and Decision Sciences 39 (2014), pp. 79-105.[64]*

12. *A. Villmann, M. Lange-Geisler, T.Villmann,* "About Semi-Inner Products for **p-QR**-Matrix Norms", *Machine Learning Reports, MLR-03-2018 (2018).[136]*

Further Publications

13. Advances in Self-Organizing Maps and Learning Vector Quantization: *Proceedings of 10th International Workshop WSOM 2014, Mittweida, Springer, Autoren: Villmann, T.; Schleif, F.-M.; Kaden, M. & Lange, M. (Eds.)*

14. *L. Fischer, M. Lange, M. Kästner, and T. Villmann,* "Accelerated vector quantization by pulsing neural gas", *Machine Learning Reports , 6(MLR-04-2012): 57–66, 2012.*

15. *T. Geweniger, M. Kästner, M. Lange, and T. Villmann*, "Derivation of a generalized Conn-index for fuzzy clustering validation", *Machine Learning Reports, 5(MLR- 07-2011):1–12, 2011.*

16. *T. Geweniger, M. Kästner, M. Lange, and T. Villmann*," Modified CONN-index for the evaluation of fuzzy clusterings" , *In M. Verleysen, editor, Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2012), pages 465–470, Louvain-La-Neuve, Belgium, 2012.*

17. *T. Geweniger, L. Fischer, M. Kaden, M. Lange, and T. Villmann*", Clustering by fuzzy neural gas and evaluation of fuzzy clusters", *Computational Intelligence and Neuroscience, 2013.*

18. *M. Kästner, M. Lange, and T. Villmann*, "Fuzzy supervised self-organizing map for semi-supervised vector quantization", *In L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh, and J.M. Zurada, editors, Artificial Intelligence and Soft Computing - Proc. the International Conference ICAISC, Zakopane , volume 1 of LNAI 7267, pages 256–265, Berlin Heidelberg, 2012. Springer.*

19. *M. Kästner, M. Strickert, D. Labudde, M. Lange, S. Haase, and T. Villmann*", Utilization of correlation measures in vector quantization for analysis of gene expression data - a review of recent developments", *Machine Learning Reports , 6(MLR-04-2012):5–22, 2012.*

20. *M. Lange and T. Villmann,* "Derivatives of l_p-norms and their approximations", *Machine Learning Reports, 7(MLR-04-2013):43–59, 2013.*

21. *M. Lange and T. Villmann*, "Partial mutual information for vector quantization", *In T. Villmann, F.-M. Schleif, M. Kaden, and M. Lange, editors, Advances in Self-Organizing Maps: 10th International Workshop WSOM 2014 Mittweida, Advances in Intelligent Systems and Computing, Berlin, 2014. Springer.*

22. *M. Lange, M. Kästner, and T. Villmann*, "About analysis and robust classification of searchlight fMRI data using machine learning classifiers", *In Proceedings of International Joint Conference on Neural Networks, Dallas, Texas, USA , pages 2026– 2033. IEEE Press, 2013.*

23. *T. Villmann, T. Geweniger, M. Kästner, and M. Lange*, "Theory of fuzzy neural gas for unsupervised vector quantization", *Machine Learning Reports, 5(MLR-06- 2011):27–46, 2011.*

24. *T. Villmann, T. Geweniger, M. Kästner, and M. Lange*, "Fuzzy neural gas for unsupervised vector quantization", *In L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh, and J.M. Zurada, editors, Artificial Intelligence and Soft Computing - Proc. the International Conference ICAISC, Zakopane, volume 1 of LNAI 7267, pages 350–358, Berlin Heidelberg, 2012 Springer.*

25. *T. Villmann, M. Kästner, and M. Lange*," Theory of patch clustering for variants of fuzzy c-means, fuzzy neural gas, and fuzzy self-organizing map", *Machine Learning Reports, 6(MLR-01-2012):80–90, 2012b.*

Nederlandse samenvatting

Dit proefschrift geeft een verenigd en veralgemeniseerd schema voor Hebbiaanse methodes in niet-Euclidische ruimtes voor ongesuperviseerd- en gesuperviseerd leren. De Hebbiaanse leermethoden worden hier onderverdeeld in vertor- en matrixmethoden.

Hebbiaanse leermethodes voor vectormethodes Over het algemeen gebruiken Hebbiaanse leermethoden, zoals het Hebbiaans PCA leren, het Euclidische inwendig product voor het behandelen van data. In dit proefschrift werden Hebbiaanse methoden die gebaseerd zijn op veralgemeniseerde inwendige producten gedefinieerd waarbij de nadruk lag op Hebbiaanse methoden voor ongesuperviseerde PCA en ICA. Het uitgangspunt was de vervanging van het Euclidische inwendige product door een algemener SIP om de Hebbiaanse methoden uit te rusten met een niet-Euclidische metriek. Daartoe werd er in hoofdstuk 3 een kader voor metriek gegeven, die door normen, gegenereerd door SIPs, geïnduceerd zijn. Deze SIPs zijn het natuurlijke equivalent van de inwendige producten voor Banach ruimten. Er werd specifiek gekeken naar bekende voorbeelden van Banach ruimtes, dat wil zeggen l_p-ruimtes en l_p-Sobolev-ruimtes met de bijbehorende SIPs in beschouwing genomen, waarbij de bijbehorende SIP voor de l_p-Sobolev-ruimte werd voorgesteld. Verder wordt er ook aandacht besteed aan inwendige producten en SIPs in algemene kernelruimtes, die de toepassing van kernelmetriek met onderliggende RKHS en RKBS mogelijk maken.

De SIPs kunnen rechtstreeks gevoerd worden aan de bovengenoemde Hebbiaanse leermethodes. Het theoretische kader voor niet-Euclidische PCA en ICA door Hebbiaans leren werd gegeven in hoofdstuk 4. De resulterende methodes zijn adaptieve PCA door Hebbiaans leren voor algemene eindig-dimensionale Banach- en Hilbert ruimtes en ook Kernel PCA door Hebbiaans leren voor RKHS en RKBS. Door het veralgemeniseren van het originele Hebbiaanse PCA leren op deze manier, wordt de kloof tussen kernelgebaseerd leren en adequate datavisualisatie gedicht wanneer kernel-leren

uitgevoerd wordt met differentieerbare kernels die prototypeleren toestaan met differentieerbare kernelmetriek in de dataruimte. Dit is ook van toepassing op andere niet-Euclidische PCA varianten, die gebaseerd zijn op l_p-normen en Sobolev-normen. Verder werd het idee van Hebbiaans leren voor kernel-PCA in niet-Euclidische ruimtes overgeheveld naar Hebbiaans ICA leren, wat leidt tot een niet-Euclidische ICA. De voorgestelde leerregel veronderstelt "whitened" data en het is belangrijk dat "prewhitening" uitgerust moet worden met de respectieve kernel metriek om de stabiliteit van het ICA-algoritme te garanderen. Daarom is Kernel PCA door Hebbiaans leren in RKHS een geschikte voorbewerkingsstap voor kernel ICA door Hebbiaans leren in RKHS. Aan het einde van hoofdstuk 4 hebben voorbeeldtoepassingen aangetoond dat deze niet-Euclidische variant van Hebbiaanse ICA in staat is niet-lineaire gemengde signalen te extraheren. Aan het einde van hoofdstuk 4 worden voorbeeldtoepassingen en simulaties van niet-Euclidische PCA getoond en de resultaten worden vergeleken met die van de Euclidische PCA. De voorbeeldtoepassingen voor PCA en ICA tonen aan dat, afhankelijk van de data en de toepassing, Hebbiaanse leermethodes die gebaseerd zijn op een algemeen inwendig product geschikter kunnen zijn voor data behandeling dan Hebbiaanse leermethodes die gebaseerd zijn op het standaard Euclidische inwendig product.

De LVQ algoritmes, die ook onder de Hebbiaans-achtige leermethodes vallen, worden behandeld in hoofdstuk 5 voor niet-Euclidische ruimtes. Een niet-Euclidische variant van GLVQ met l_p-normen vereist de afgeleiden van dezen, die niet mogelijk zijn bij de oorsprong ($x = 0$) wegens de absolutewaardefunctie in de l_p-normen. Daarom worden twee gladde benaderingen van de maximumfunctie, respectievelijk de α-softmax en de α-quasimax functies, en de bijbehorende afgeleiden bestudeerd. Het gebruik van de α-quasimax functie voor het benaderen van de maximumfunctie is aan te raden. Verder worden gladde benaderingen van de absolutewaardefunctie, die gebaseerd zijn op de maximumfunctie, ook bestudeerd en tevens de bijbehorende afgeleiden. Over het algemeen moet worden opgemerkt dat de onderliggende benaderingsfuncties voor l_p-normen die toegepast worden in GLVQ, waaronder de afgeleiden van de maximumfunctie en de absolutewaardefunctie, consistent moeten zijn. De toepassingen aan het einde van hoofdstuk 5 hebben aangetoond dat op l_p-normen gebaseerde afstanden succesvol toegepast kunnen worden door het gebruik van de benaderingsfuncties. Ook verbetert een consistente methode in GLVQ de classificatieprestatie ten opzichte van een inconsistente variant.

Hebbiaanse leermethodes voor matrixmethodes Een ander hoofdonderdeel van dit proefschrift was Hebbiaanse methoden in niet-Euclidische ruimtes voor ma-

trixen. Daarvoor worden in het tweede deel van hoofdstuk 3 de Banach ruimtes van matrixen met de bijbehorende SIPs bestudeerd, waarbij de focus lag op de Schatten-p-normen. Een SIP voor deze Banach ruimte die de respectieve norm genereert werd ontwikkeld. De **QR**-norm, die beschouwd kan worden als een uitgebreidere variant van de Schatten-p-norm, werd nader bestudeerd omtrent de eigenschappen van de norm, dat wil zeggen dat de eigenschappen die geïntroduceerd zijn door G. Allen in [4], aangepast worden. Verder wordt er een inwendig product voor deze Hilbert ruimte, die de **QR**-norm genereert, voorgesteld. Toch blijft de ontwikkeling van een SIP voor de **QR**-norm met willekeurige $p \neq 2$ een open probleem.

De resulterende (semi)-inwendige producten kunnen rechtstreeks toegepast worden in de Hebbiaanse PCA methode voor matrixen, voor welke de theorie en het bewijs van de vereiste eigenschappen gegeven werden aan het einde van hoofdstuk 4. De resulterende methode is een Hebbiaanse leerregel die gebaseerd is op Schatten-p-norms voor principiële componenten in Banach ruimtes van matrixen. In het laatste deel van hoofdstuk 4 werd het verschillende gedrag van de matrixmethode vergeleken met de vectorvariant getoond voor een illustratief voorbeeld met het resultaat dat in het geval waarin matrixdata, dat belangrijke ruimtelijke informatie bevat, wordt behandeld, het gebruik van de matrix PCA tot een betere prestatie leidt.

Verder kunnen matrix normen worden gebruikt als ongelijkheidsmaat voor gesuperviseerde Hebbiaans-achtige leermethodes. In hoofdstuk 5 werd de LVQ methode uitgebreid voor de classificatie van matrix data met Schatten-p-normen en dit wordt learning matrix quantization (LMQ) genoemd. Vergeleken met de vertor-tegenhanger leidt het gebruik van matrixnormen tot een betere structurele flexibiliteit van relevantieleren in GLMQ. De resulterende methoden, die gebaseerd zijn op verscheidene soorten van algebraïsche relevantiewegingen, zijn GLMQ met Hadamard-Relevance-Learning (GLMQ_{HRL}) en left/right Multiplicative-Relevance-Learning ($\text{GLMQ}_{\text{l/r-MLR}}$). Naast de Schatten-$p$-norm werd ook de **QR**-norm toegepast in GLMQ en hiernaar wordt gerefereerd als QR-Relevance-Learning (GLMQ_{QR}). Het Kronecker product tussen matrixen en tensorcomposieten met een relevantietensor in plaats van een relevantiematrix brengt verdere soorten van algebraïsche composities in GLMQ voort, die respectievelijk worden aangeduid met Kronecker-Relevance-Learning (KRL) en Tensor-Relevance-Learning (TRL). Het verkrijgen van een tegenhanger van GMLVQ was een reden voor het toepassen van deze composities. De verwachte variatie van KRL in GLMQ ontaardt in één parameter en derhalve is GLMQ_{KRL} nutteloos. Een vergelijkbaar resultaat werd verkregen voor GLMQ met relevantietensors van de vierde orde, dat wil zeggen dat de variatie in het relevantieleren hetzelfde is als bij GLMQ_{KRL}. In het geval waarin relevantietensors van de derde orde toegepast wer-

den, resulteerde de tensoroperatie in een vector en daarom is deze variant ook geen tegenhanger van GMLVQ.

Samengevat is de betere flexibiliteit van het relevantieleren het voordeel van LMQ ten opzichte van LVQ, maar het is doorgaans ingewikkelder en over het algemeen moeilijk te interpreteren. Dit werd duidelijk door de voorbeeldtoepassingen aan het einde van hoofdstuk 5.

Bibliography

[1] R. A. Adams and J. F. Fournier. *Sobolev Spaces*. Pure and Applied Mathematics. Elsevier Science, 2 edition, 2003.

[2] HH. Alahmadi, Y. Shen, S. Fouad, C.D. Luft, P. Bentham, Z. Kourtzi, and P. Tino. Classifying cognitive profiles using machine learning with privileged information in mild cognitive impairment. *Frontiers in Computational Neuroscience*, 10(117), 2016.

[3] J.L. Alba, A. Pujol, and J.J. Villanueva. Separating geometry from texture to improve face analysis. *IEEE International Conference on Image Processing*, 2: 673–676, 2001.

[4] G. I. Allen, L. Grosenick, and J. Taylor. A generalized least-square matrix decomposition. *Journal of the American Statistical Association*, 109:145–159, 2014.

[5] S. Amari, A. Cichocki, and H. H. Yang. A new learning algorithm for blind signal separation. In *Advances in Neural Information Processing Systems*, pages 757–763. MIT Press, 1996.

[6] N. Aronszajn. Theory of reproducing kernels. *Transaction of the America Mathematical Society*, 68(337-404), 1950.

[7] A. Asuncion and D. Newman. Indian diabetes data set (pima). *http://archieve.ics.uci.edu/ml/*.

[8] F.R. Bach and M.I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

[9] G. H. Ball and D. J. Hall. A clustering technique for summarizing multivariate data. *Behavioral Science*, 12(2):153–155, 1967.

[10] A. J. Bell and T. J. Sejnowski. An information maximisation approach to blind separation and blind deconvolution. *Neural Computation*, 6:1126–1159, 1995.

[11] Y. Bengio. Learning deep architectures for ai. *Foundation and Trends in Machine Learning*, 2(1):1–127, 2009.

[12] Y. Bengio. *Neural Networks: Tricks of the Trade*, chapter Practical Recommandations for Gradient-based Training of Deep Architecturs, pages 437–478. Springer, Berlin Heidelberg, 2012.

[13] M. Biehl, R. Breitling, and Y. Li. Analysis of tilling microarray data by learning vector quantization and relevance learning. In H. Yin, P. Tino, E. Corchado, W. Byrne, and X. Yao, editors, *Proc. Intelligent Data Engineering and Automated Learning (IDEAL)*, number 4881 in LNCS, pages 880–889. Springer, 2007.

[14] M. Biehl, M. Kästner, M. Lange, and T. Villmann. Non-Euclidean principal component analysis and Oja's learning rule - theoretical aspects. In P.A. Estevez, J.C. Principe, and P. Zegers, editors, *Advances in Self-Organizing Maps: 9th International Workshop WSOM 2012 Santiage de Chile*, volume 198 of *Advances in Intelligent Systems and Computing*, pages 23–34, Berlin, 2013. Springer.

[15] M. Biehl, B. Hammer, F.-M. Schleif, P. Schneider, and T. Villmann. Stationarity of matrix relevance lvq. *International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2015.

[16] E. Bingham and A. Hyvärinen. A fast fixed-point algorithm for independent component analysis of complex valued signals. *Int. J. of Neural Systems*, 10(1): 1–8, 2000.

[17] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer Science+Business Media, LLC, New York, NY, 2006.

[18] A. Bohnsack, K. Domaschke, M. Kaden, M. Lange, and T. Villmann. Mathematical characterization of sophisticated variants for relevance learning in learning matrix quantization based on schatten-p-norms. *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, 1:403–414, 2015.

[19] A. Bohnsack, K. Domaschke, M. Kaden, M. Lange, and T. Villmann. Learning matrix quantization and relevance learning based on schatten-p-norms. *Neurocomputing*, 192:104–114, 2016.

[20] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press,, 2004.

[21] K. Bunte, P. Schneider, B. Hammer, F.-M. Schleif, T. Villman, and M. Biehl. Limited rank matrix learning, discriminative dimension reduction and visualization. *Neural Networks*, 26:159–173, 2012.

[22] Deng Cai, Xiaofei He, Jiawei Han, and Hongjiang Zhang. Orthogonal laplacianfaces for face recognition. *IEEE Transactions on Image Processing*, 15: 3608–3614, 2006.

[23] N. W. Campbell, B. T. Thomas, and T. Troscianko. Automatic segmentation and classification of outdoor images using neural networks. *International Journal of Neural Systems*, 8(1):137–144, 1997.

[24] J. C. W. Chan, K. P. Chan, and A. G. O. Yeh. Detecting the nature of change in an urban environment: A comparison of machine learning algorithms. *Photogrammetric Engineering and Remote Sensing*, 67(2):213–225, February 2001.

[25] C. Chen and O.L. Mangasarian. Smoothing methods for convex inequalities and linear complementarity problems. *Mathematical Programming*, 71(1):51–69, 1995. ISSN 0025-5610. doi: 10.1007/BF01592244.

[26] A. Cichocki and S. Amari. *Adaptive Blind Signal and Image Processing*. John Wiley, 2002.

[27] P.L. Clark, Z.-P. Liu, J. Zhang, and L. M. Gierasch. Intrinsic tryptophans of crabpi as probes of structure and folding. *Protein Science*, 5(6):1108–1117, 1996.

[28] P. Comon and C. Jutten. *Handbook of Blind Source Separation*. Academic Press, 2010.

[29] J. Cook. Basic properties of the soft maximum. Working Paper Series 70, UT MD Anderson Cancer Center Department of Biostatistics, 2011. http://biostats.bepress.com/mdandersonbiostat/paper70.

[30] T. Cover and P. Hart. Nearest neighbor pattern classication. *IEEE Transactionson Information Theory*, 13:21–27, 1967.

[31] M. Day. The spaces L_p with $0 < p < 1$. *Bulletin of the American Mathematical Society*, 46:816–823, 1940.

[32] K. Domaschke, M. Kaden, M. Lange, and T. Villmann. Learning matrix quantization and variants of relevance learning. In M. Verleysen, editor, *Proceesings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2015)*, pages 13–18, Louvain-La-Neuve, Belgium, 2015.

[33] O. Holz et al. Volatile organic compounds (voc) in exhaled breath after experimental ozone exposure. *Proc. of the American Thoracic Society 2013 International Conference*, 2013.

[34] A.S. Georghiades. Yale face data set. URL `http://archive.ics.Uci.edu/ml/`.

[35] J. Giles. Classes of semi-inner product spaces. *Transaction of the America Mathematical Society*, (129):436–446, 1967.

[36] K.-H. Goldhorn, H.-P. Heinz, and M. Kraus. *Moderene mathematische Methoden der Physik*. Springer-Verlag, Berlin-Heidelberg, 2009.

[37] O. Golubitski and S.M. Watt. Distance-based classification of handwritten symbols. *International Journal on Document Analysis and Recognition (IJDAR)*, 13(2):133–146, 2010.

[38] Z. Gu, M. Shao, L. Li, and Y. Fu. Discriminative metric: Schatten norms vs. vector norm. In *Proc. of The 21st International Conference on Pattern Recognition (ICPR 2012)*, pages 1213–1216, 2012.

[39] S. Günter, N.N. Schraudolph, and S.V.N Vishwanathan. Fast iterative kernel principal component analysis. *Journal of Machine Learning Research*, 8:1893–1918, 2007.

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

[41] O. Hanner. On the uniform convexity of L_p and l_p. *Arkiv för Matematik*, 3(19): 239–244, 1956.

[42] S. Harmeling, A. Ziehe, and M. Kawanabe and. K.-R. Müller. Kernel-based nonlinear blind source separation. *Neural Computation*, 15(5):1089–1124, 2003.

[43] C. Harth. Erweiterung von geralized [relevance, matrix] learning vector quantization zur anwendung auf funktionelle daten. Master's thesis, University of Applied Sciences Mittweida, Saxony, Germany, 2012.

[44] M. H. Hassoun. *Fundamentals of Artificial Neural Networks*. MIT Press, 1995.

[45] S. Hayken. *Neural Networks and Learning Machines*. Number Bd. 10 in Neural Networks and Learning Machines. Prentice Hall, 3 edition, 2009.

[46] Simon Haykin. *Neural Networks - A Comprehensive Foundation*. IEEE Press, New York, 1994.

[47] D. O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. Wiley and Sons, 1949.

[48] J. A. Hertz, A. S. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation*. Westview Press, 1991.

[49] T. Hoffmann, B. Schölkopf, and A.J. Smola. Kernel methods in machine learning. *The Annals of Statistics*, 36(3):1171–1220, 2008.

[50] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, The Johns Hopkins University, 2nd edition, 2013.

[51] A. G. Horvath. Semi-infenite inner product and generalized minkowski spaces. *Journal of Geometry and Physics*, 60:1190–1208, 2010.

[52] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Education Psychology*, 24:417–441 and 498–520, 1933.

[53] A. Hyvaerien, J. Karhunen, and E. Oja. *Independent Component Analysis*. A Wiley-Interscience Publication, 2001.

[54] A. Hyvärinen and E. Oja. Simple neuron models for independent component analysis. *International Journal of Neural Systems*, 6:671–687, 1997.

[55] A. Hyvärinen and E. Oja. Independent component analysis by general nonlinear hebbian-like learning rules. *Signal Processing*, 64:301–313, 1998.

[56] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. J. Wiley & Sons, 2001.

[57] Y. Bengio I. Goodfellow and A. Courville. *Deep Learning*. MIT Press, 2016.

[58] I.Kantorowitsch and G. Akilow. *Funktionalanalysis in normierten Räumen*, volume 2. Akademie-Verlag, Berlin, 1978.

[59] H. Itoh, A. Imiya, and T. Sakai. Low-dimensional tensor prinicipal component analysis. In G. Azzopardi and N. Petkov, editors, *Computer Analysis of Images and Patterns*, volume Part II, pages 715–726. Berlin-Heidelberg Springer, 2015.

[60] P.K. Jain and K. Ahmad. Unconditional Schauder basis and best approximations in Banach spaces. *Indian Journal of Pure and Applied Mathematics*, 12 (12):1456–1467, 1981.

[61] P.K. Jain and K. Ahmad. Schauder decomposition and best approximations in Banach spaces. *Portugaliae Mathematica*, 44(1):25–39, 1987.

[62] R. James. Bases in banach spaces. *American Mathematical Monthly*, 89:625–640, 1982.

[63] C. Jutten and J. Karhunen. Advances in blind source separation (BSS) and independent component analysis (ICA) for nonlinear mixtures. *International Journal of Neural Systems*, 14(5):267–292, 2004.

[64] M. Kaden, M. Lange, D. Nebel, M. Riedel, T. Geweniger, and T. Villmann. Aspects in classification learning - Review of recent developments in Learning Vector Quantization. *Foundations of Computing and Decision Sciences*, 39: 79–105, 2014.

[65] M. Kaden, M. Riedel, W. Hermann, and T. Villmann. Border sensitive learning in generalized learning vector quantization: As an alternative to support vector machines. *Soft Computing*, 19(9):2423–2434, 2015.

[66] M. Kästner, B. Hammer, M. Biehl, and T. Villmann. Functional relevance learning in generalized learning vector quantization. *Neurocomputing*, 90(9): 85–95, 2012.

[67] M. Kästner, D. Nebel, M. Riedel, M. Biehl, and T. Villmann. Differentiable kernels in generalized matrix learning vector quantization. In *Proc. of the Internacional Conference of Machine Learning Applications (ICMLA'12)*, pages 1–6. IEEE Computer Society Press, 2012.

[68] G. A. Khuwaja. An adaptive combined classifier system for invariant face recognition. *Digital Signal Processing: A Review Journal*, 12,(1):21– 46, January 2002.

[69] K.I. Kim, M.O. Franz, and B. Schölkopf. Kernel hebbian algorithm for iterative kernel principal component analysis. Technical Report 109, Max-Planck-Institute for Biological Cybernetics, June 2003.

[70] K.I. Kim, M.O. Franz, and B. Schölkopf. Iterative kernel principal component analysis for image modelling. *IEEE Transactions on Pat*, 27(9):1351–1366, 2005.

[71] T. Kohonen. Learning vector quantization for pattern recognition. Technical Report TKKF-A601, Laboratory of Computer and Information Science, Department of Technical Physics, Helsinki University of Technology, Helsinki, Finland, 1986.

[72] T. Kohonen. Learning vector quantization. *Neural Networks*, 1, 1988.

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

[74] T. Kohonen. *Self-Organizing Maps*, volume 3 of *Information Sciences*. Springer, 2001.

[75] A.N. Kolmogorov and S.V. Fomin. *Reelle Funktionen und Funktionalanalysis*. VEB Deutscher Verlag der Wissenschaften, Berlin, 1975.

[76] R. Kress. *Numerical Analysis*, volume 181. Springer New York, 1998.

[77] H.J. Kushner and D.S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer-Verlag, New York, 1978.

[78] J.R. Lakowicz. *Principles of Fluorescence Spectroscopy*. Springer, 3rd edition, 2006.

[79] M. Lange and T. Villmann. Derivatives of l_p-norms and their approximations. *Machine Learning Reports*, 7(MLR-04-2013):43–59, 2013.

[80] M. Lange, M. Biehl, and T. Villmann. Non-Euclidean independent component analysis and Ojaś learning. In M. Verleysen, editor, *Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2013)*, pages 125–130, Louvain-La-Neuve, Belgium, 2013. i6doc.com.

[81] M. Lange, D. Nebel, and T. Villmann. Non-Euclidean principal component analysis for matrices by Hebbian learning. In L. Rutkowski, M. Korytkowski,

R. Scherer, R. Tadeusiewicz, L.A. Zadeh, and J.M. Zurada, editors, *Artificial Intelligence and Soft Computing - Proc. the International Conference ICAISC*, volume 8467 of *Lecture Notes in Computer Science*, pages 77–88, Zakopane, 2014. Springer.

[82] M. Lange, D. Zühlke, O. Holz, and T. Villmann. Applications of l_p-norms and their smooth approximations for gradient based learning vector quantization. In M. Verleysen, editor, *Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2014)*, pages 271–276, Louvain-La-Neuve, Belgium, 2014.

[83] M. Lange, M. Biehl, and T. Villmann. Non-euclidean principal component analysis by hebbian learning. *Neurocomputing*, 147:107–119, 2015.

[84] J. Lee and M. Verleysen. Generalization of the l_p norm for time series and its application to self-organizing maps. In M. Cottrell, editor, *Proc. of Workshop on Self-Organizing Maps (WSOM) 2005*, pages 733–740, Paris, Sorbonne, 2005.

[85] D. Liu, S. Hu, and J. Wang. Global output convergence of a class of continuous-time recurrent neural networks with time-varying thresholds. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 51(4):161–167, April 2004.

[86] S. Liu and G. Trenkler. Hadamard, khatri-rao, kronecker and other matrix products. *International Journal of Information and System Sciences*, 4(1):160–177, 2008.

[87] Gabriele Lohmann, Johannes Stelzer, Jane Neumann, Nihat Ay, and Robert Turner. "more is different "in functional magnetic resonance imaging: A review of recent data analysis techniques. *Brain Connectivity*, 3(3), 2013.

[88] G. Lumer. Semi-inner-product spaces. *Transaction of the America Mathematical Society*, (100):29–43, 1961.

[89] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten. Neuralgas network for vector quantization and its application to timeseries prediction. *IEEE Transactions on Neural Networks*, 4(4):558 – 569, 1993.

[90] D. Martinez and A. Bray. Nonlinear blind source separation using kernels. *IEEE Transactions on Neural Networks*, 14(1):228–235, 2003.

[91] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society, London, A*, 209:415–446, 1909.

[92] B. Nath. On a generalization of semi-inner product spaces. *Mathematical Journal of Okayama University*, 1(15):1–6, 1971.

[93] B. Nath. Topologies on generalized semi-inner product spaces. *Composito Mathematica*, 3(23):309–316, 1971.

[94] D. Nebel, B. Hammer, K. Frohberg, and T. Villmann. Median variants of learning vector quantization for learning of dissimilarity data. *Neurocomputing*, 169:295–305, 2015.

[95] D. Nova and P.A. Estévez. A review of learning vector quantization classifiers. *Neural Computation and Applications*, 2013. doi: 10.1007/s00521-013-1535-3. URL `http://dx.doi.org/10.1007/s00521-013-1535-3`.

[96] E. Oja. A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15:267–273, 1982.

[97] E. Oja. Neural networks, principle components and subspaces. *International Journal of Neural Systems*, 1:61–68, 1989.

[98] E. Oja. Nonlinear pca: Algorithms and applications. In *The World Congress on Neural Networks Portland*, pages 396–400, 1993. Portland.

[99] S. Pattanayak. *Pro Deep Learning with TensorFlow: A Mathematical Approach to Advanced Artificial Intelligence in Python*. Apress, 2017.

[100] E. Pekalska and R.P.W. Duin. *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications*. World Scientific, 2006.

[101] D. Peng and Z. Yi. A modified oja xu mca learning algorithm and its convergence analysis. *IEEE Transactions on circuits and systems*, 54:348–352, 2007.

[102] S. V. Phadke and N. K. Thakare. Projection operators on uniformly convex semi-inner product spaces. *Indian National Science Acadamy Journals*, 7(12): 1438–1447, 1974.

[103] D.T. Pham, P. Garat, and C. Jutten. Separation of a mixture of independent sources through a maximum likelihood approach. In *Proceedings of EUSIPCO*, pages 771–774, 1992.

[104] B. Póczos, S. Kirshner, and C. Szepesvári. REGO: Rank based estimation of Rényi information using Euclidean graph optimization. In *Proc. of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9 of Journal of Machine Learning Research (JMLR), 2010.

[105] J.O. Ramsay and B.W. Silverman. *Functional Data Analysis*. Springer Science+Media, New York, 2nd edition, 2006.

[106] S. C. Rastogi. *Cell And Molecular Biology*. New Age International, 2006.

[107] J.R. Retherford and R.C. James. Unconditional bases and best approximation in Banach spaces. *Bulletin of the American Mathematical Society*, 75(1):108–112, 1969.

[108] M. Riedel, F. Rossi, M. Kästner, and T. Villmann. Regularization in relevance learning vector quantization using l_1-norms. In M. Verleysen, editor, *Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2013)*, pages 17–22, Louvain-La-Neuve, Belgium, 2013. i6doc.com.

[109] F. Riesz and B. Sz.-Nagy. *Vorlesungen über Funktionalanalysis*. Verlag Harri Deutsch, 4 edition, 1982.

[110] H. Ritter, T. Martinetz, and K. Schulten. *Neuronale Netze, Eine Einführung in die Neuroinformatik selbstorganisierender Netzwerke*. Addison Wesley, 1990.

[111] F. Rosenblatt. *Principle of Neurodynamics; Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, 1962.

[112] F. Rossi, B. Conan-Guez, and A. El Golli. Clustering functional data with the SOM algorithm. In M.Verleysen, editor, *European Symposium on Artificial Neural Networks 2004*, pages 305–312. d-side publications, 2004.

[113] F. Rossi, N. Delannay, B. Conan-Gueza, and M. Verleysen. Representation of functional data in neural networks. *Neurocomputing*, 64:183–210, 2005.

[114] L. Sachs. *Angewandte Statistik*. Springer Verlag, 7 edition, 1992.

[115] T. D. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, 2:459–473, 1989.

[116] A. Sato and K. Yamada. Generalized learning vector quantization. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 423–429, Cambridge, MA, USA, 1996. MIT Press.

[117] R. Schatten. *A Theory of Cross-Spaces*, volume 26 of *Annals of Mathematics Studies*. Princeton University Press, 1950.

[118] M. Schmidt, G. Fung, and R. Rosales. Fast optimization methods for l_1 regularization: A comparative study and two new approaches. In J.M. Kok, J. Koronacki, R.L. Mantaras, S. Matwin, D. Mladenič, and A. Skowron, editors, *Machine Learning: ECML 2007*, volume 4701 of *Lecture Notes in Computer Science*, chapter 28, pages 286–297. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

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

[120] P. Schneider, K. Bunte, H. Stiekema, B. Hammer, and T. Villmann M. Biehl. Regularization in matrix relevance learning. *IEEE Transactions on Neural Networks*, 21(5):831–840, 2010.

[121] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

[122] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neur*, 14(7):1299–1319, 1998.

[123] D. B. Siano and D. E. Metzler. Band shapes of the electronic spectra of complex molecules. *The Journal of Chemical Physics*, 51(5):1856–1861, 1969.

[124] H. Sompolinsky. *The Theory of Neural Networks: The Hebb Rule and Beyond*, volume 275 of *Lecture Notes in Physics*. Springer Berlin Heidelberg, 1987.

[125] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis and Machine Vision*. Brooks Publishing, 2nd edition, 1998.

[126] M. Soriano, L. Garcia, and C. Saloma. Fluorescent image classification by major color histograms and a neural network. *Optics Express*, 8(5):271–277, February 26 2001.

[127] James C. Spall. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. John Wiley & Sons, 2005.

[128] Suvrit Sra, Sebastian Nowozin, and Stephen J. Wright. *Optimization for Machine Learning*. Neural Information Processing series. The MIT Press, 2011.

[129] S. Stalke. *Femtosekunden Pump ProbeAbsorptionsspektroskopie zur Untersuchung der intramolekularen Dynamik von beta Apo Carotinsaeuren und von Patman in verschieden Loesungsmitteln*. PhD thesis, Georg-August-Universitaet Goettingen, 2011.

[130] I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2001.

[131] I. Steinwart and A. Christmann. *Support Vector Machines (Information Science and Statistics)*. Springer, 2008.

[132] I. Steinwart, D. Hush, and C. Scovel. An explicit description of the reproducing kernel hilbert spaces of gaussian rbf kernels. *IEEE Transactions on Information Theory*, 52(10):4635–4643, 2006.

[133] J. V. Stone. *Independent Component Analysis, A Tutorial Introduction*. A Bradford Book, 2004.

[134] R. S. Sutton and A. G. Barto. Reinforcement learning: An introdcuction. In *IEEE Transactions on Neural Networks*, volume 5, pages 1054–1054, 1998.

[135] H. Thodberg. Tecator meat sample dataset. *http://lib.stat.cmu.edu/datasets/tecator*.

[136] A. Villmann, M. Lange-Geisler, and T. Villmann. About semi-inner products for $\boldsymbol{p}$-$\boldsymbol{QR}$-matrix norms. Machine Learning Report 3, 2018.

[137] T. Villmann. Sobolev metrics for learning of functional data. *Machine Learning Reports*, MLR-03-2007(1):1–15, 2007. ISSN 1865-3960.

[138] T. Villmann and S. Haase. A note on gradient based learning in vector quantization using differentiable kernels for Hilbert and Banach spaces. Technical Report MLR-02-2012, 2012.

[139] T. Villmann and B. Hammer. Functional principal component learning using oja's method and sobolev norms. In R. Miikkulaien J. Principe, editor, *Advances in Self Organizing Maps, Proceeding of the Workshop on Self Organizing Maps*, pages 325–333. Springer, 2009.

[140] T. Villmann and M. Lange. A comment on the functional $\boldsymbol{L_p^{TS}}$-measure regarding the norm properties. Machine Learning Reports 02, 2015.

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

[142] T. Villmann, M. Kästner, D. Nebel, and M. Riedel. ICMLA face recognition challenge – results of the team 'Computational Intelligence Mittweida'. In *Proc. of the Internacional Conference of Machine Learning Applications (ICMLA'12)*, pages 7–10. IEEE Computer Society Press, 2012.

[143] T. Villmann, M. Kästner, D. Nebel, and M. Riedel. Lateral enhancement in adaptative metric learning for functional data. *Neurocomputing*, 131:23–31, 2014.

[144] L. Xu, E. Oja, and C. Y. Suen. Modified hebbian learning for curve and surface fitting. *Neural Networks*, 5:441–457, 1992.

[145] A. L. Yuille, D. M. Kammen, and D. S. Cohen. Quadrature and the development of orientation selective cortical cells by hebb rules. *Biological Cybernetics*, 61: 183–194, 1989.

[146] H. Zhang and J. Zhang. Generalized semi-inner products with applications to regularized learning. *Journal of Mathematical Analysis and Application*, (372): 181–196, 2010.

[147] H. Zhang, Y. Xu, and J. Zhang. Reproducing kernel banach spaces for machine learning. *Journal of Machine Learning Research*, 10:2741–2775, 2009.

[148] Q. Zhang and Y.-W. Leung. Energy function for the one-unit oja algorithm. *IEEE Transactions on Neural Networks*, 6(5):1291–1293, 1995.

Hebbian Learning Approaches based on
General Inner Products and Distance Measures in Non-Euclidean Spaces

Mandy Lange-Geisler

ISBN: 978-94-034-1470-6 (printed version)
ISBN: 978-94-034-1469-0 (electronic version)