Supervised & Unsupervised Transfer Learning

Inauguraldissertation

zur
Erlangung der Würde eines Doktors der Philosophie
vorgelegt der
Philosophisch-Naturwissenschaftlichen Fakultät
der Universität Basel

von

Julia E. Vogt

aus Deutschland

Basel, 2013

creative commons

Namensnennung-Keine kommerzielle Nutzung-Keine Bearbeitung 2.5 Schweiz

---

Sie dürfen:

das Werk vervielfältigen, verbreiten und öffentlich zugänglich machen

Zu den folgenden Bedingungen:

Namensnennung. Sie müssen den Namen des Autors/Rechteinhabers in der von ihm festgelegten Weise nennen (wodurch aber nicht der Eindruck entstehen darf, Sie oder die Nutzung des Werkes durch Sie würden entlohnt).

Keine kommerzielle Nutzung. Dieses Werk darf nicht für kommerzielle Zwecke verwendet werden.

Keine Bearbeitung. Dieses Werk darf nicht bearbeitet oder in anderer Weise verändert werden.

- Im Falle einer Verbreitung müssen Sie anderen die Lizenzbedingungen, unter welche dieses Werk fällt, mitteilen. Am Einfachsten ist es, einen Link auf diese Seite einzubinden.
- Jede der vorgenannten Bedingungen kann aufgehoben werden, sofern Sie die Einwilligung des Rechteinhabers dazu erhalten.
- Diese Lizenz lässt die Urheberpersönlichkeitsrechte unberührt.

Quelle: http://creativecommons.org/licenses/by-nc-nd/2.5/ch/ Datum: 3.4.2009

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät

auf Antrag von

Prof. Dr. Volker Roth, Universität Basel, Dissertationsleiter
Prof. Dr. Joachim Buhmann, ETH Zürich, Korreferent

Basel, den 11.12.2012

Prof. Dr. Jörg Schibler, Dekan

Abstract

This thesis investigates transfer learning in two areas of data analysis, supervised and unsupervised learning. We study *multi-task* learning on vectorial data in a supervised setting and *multi-view* clustering on pairwise distance data in a Bayesian unsupervised approach. The aim in both areas is to transfer knowledge over different related data sets as opposed to learning on single data sets separately.

In supervised learning, not only the input vectors but also the corresponding target vectors are observed. The aim is to learn a mapping from the input space to the target space to predict the target values for new samples. In standard classification or regression problems, one data set at a time is considered and the learning problem for every data set is solved separately. In this work, we are looking at the non-standard case of learning by exploiting the information given by multiple related tasks. Multi-task learning is based on the assumption that multiple tasks share some features or structures. One well-known technique solving multi-task problems is the Group-Lasso with 2-norm regularization. The motivation for using the Group-Lasso is to couple the individual tasks via the group-structure of the constraint term. Our main contribution in the supervised learning part consists in deriving a complete analysis of the Group-Lasso for *all* p-norm regularizations, including results about uniqueness and completeness of solutions and coupling properties of different p-norms. In addition, a highly efficient active set algorithm for all p-norms is presented which is guaranteed to converge and which is able to operate on extremely high-dimensional input spaces. For the first time, this allows a direct comparison and evaluation of all possible Group-Lasso methods for all p-norms in large scale experiments. We show that in a multi-task setting, both, tight coupling norms with $p \gg 2$ and loose coupling norms with $p \ll 2$ significantly degrade the prediction performance. Moderate coupling norms for $p \in [1.5, 2]$ seem to be the best compromise between coupling strength and robustness against systematic differences between the tasks.

The second area of data analysis we look at is unsupervised learning. In unsupervised learning, the training data consists of input vectors without any corresponding target vectors. Classical problems in unsupervised learning are clustering, density estimation or dimensionality reduction. As in the supervised scenario, we are not only considering single data sets independently of each other, but we want to learn over two or more data sets simultaneously. A problem that arises frequently is that the data is only available as pairwise distances between objects (e.g. pairwise string alignment scores from protein sequences) and a loss-free embedding into a vector space is usually not possible. We propose a Bayesian clustering model that is able to operate on this kind of distance data without explicitly embedding it into a vector space. Our main contribution in the unsupervised learning part is twofold. Firstly, we derive a fully probabilistic clustering method based on pairwise Euclidean distances, that is rotation-, translation-, and scale- invariant and uses the Wishart distribution in the likelihood term. On the algorithmic side, a highly efficient sampling algorithm is presented. Experiments indicate the advantage of encoding the translation invariance into the likelihood and our clustering algorithm clearly outperforms several hierarchical clustering methods. Secondly, we extend this clustering method to a novel Bayesian multi-view clustering approach based on distance data. We show that the multi-view clustering method reveals shared information between different views of a phenomenon and we obtain an improved clustering compared to clustering on every view separately.

Acknowledgements

This dissertation is based on my research that I carried out as a Ph.D. student at the University of Basel. During this time, I had the guidance, support, and friendship of a number of people. It is a pleasure to thank the many people who made this thesis possible.

I have been extremely fortunate to have Prof. Volker Roth as my advisor. I would like to thank him for his encouragements, constructive suggestions and constant support during this research. His door was always open to me whenever I needed his help. His open-mindedness and academic guidance as well as his contagious enthusiasm for doing research was a constant source of motivation for me. Without his sound advice and continuing support, this thesis would not have been possible. I am especially grateful for all the opportunities Prof. Roth opened up, not only by providing scientific ideas but also by encouraging interdisciplinary cooperation with researchers in the biomedical field. Prof. Roth always succeeded to guarantee a productive and cooperative environment for scientific research. I consider myself very lucky that I got the opportunity to accomplish this thesis under his guidance.

I am very grateful to my co-examiner Prof. Joachim Buhmann for reviewing my thesis. I feel honored by his interest in my work.

It was a pleasure to work with Prof. Markus Heim, Michael Dill and Zuzanna Makowska at the University Hospital Basel. I would like to thank them for an extraordinary good collaboration. This collaboration in the field of liver diseases was extremely interesting and proved a great experience and an enormous personal enrichment.

Here I wish to thank my fellow Ph.D. students Sudhir Raman, Sandhya Prabhakaran, Melanie Rey and David Adametz from the biomedical data analysis group for the many fruitful discussions and collaborations. I would also like to thank them for maintaining an open and casual research environment where ideas could be exchanged easily.

I also want to thank Nadine Fröhlich, Marcel Lüthi, Diego Milano,Ghazi Bouabene, Manolis Sifilakis, Michael Springmann and Melina Inderbitzin who enriched my time in Basel on a personal level.

I would like to express my gratitude to Alberto Giovanni Busetto, Daniel Stekhoven, Werner Kovacs, Niklaus Fankhauser, Eduard Sabido and Yibo Wu for their efforts in our joint project on hepatic insulin resistance.

Finally, I want to thank my family for their understanding, endless patience and support when it was most required. Especially I want to thank Bobo Nick who shared this journey with me. His love, encouragement and support made possible everything. Thank you!

Contents

Notations

I_n	Identity matrix of size n
$\mathbf{0}_n$	Zero vector of size n
$\mathbf{1}_n$	Vector of all-ones of size n
$\mathbb{R}$	Real numbers
$\mathbb{R}^n_+$	$\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} \geq 0\}$
$\mathbb{R}^n_{++}$	$\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} > 0\}$
X	Data matrix in $\mathbb{R}^{n \times d}$
X^T	Transpose of X in $\mathbb{R}^{d \times n}$
S	Similarity or dot-product matrix in $\mathbb{R}^{n \times n}$, $S = XX^T$
D	Distance matrix in $\mathbb{R}^{n \times n}$, $D_{ij} = S_{ii} + S_{jj} - 2S_{ij}$
B	Partition matrix in $\mathbb{R}^{n \times n}$
k_B	Number of blocks present in B
n_b	Size of block $b \in B$
$\boldsymbol{x}$	A column vector
$\boldsymbol{x}^T$	Transpose of a vector $\boldsymbol{x}$
$\text{rank}(X)$	Rank of a matrix X
$N(X)$	Nullspace of a matrix X
tr(X)	Trace of a matrix X
$\text{Diag}(\boldsymbol{x})$	$n \times n$ diagonal matrix with components of $\boldsymbol{x} \in \mathbb{R}^n$ on diagonal
$\mathcal{N}(\mu, \Sigma)$	Normal distribution with mean μ and covariance matrix Σ
$Dir(\boldsymbol{\theta})$	Dirichlet distribution with parameter vector $\boldsymbol{\theta}$
$\mathcal{W}_d(\Sigma)$	Wishart distribution with covariance matrix Σ and d degrees of freedom
$p(y\|a, b)$	Probability of y given parameters a and b
$l(\cdot)$	Likelihood function
$\mathcal{L}(\cdot, \cdot)$	Lagrangian function
ℓ_p	p-norm of a vector $\boldsymbol{x} \in \mathbb{R}^n$, $\|\boldsymbol{x}\|_p = (\sum_{i=1}^n \|x_i\|^p)^{\frac{1}{p}}$
$\ell_{1,p}$	Sum of ℓ_p-norms of sub-vectors $\boldsymbol{x}_j$, $\sum_{j=1}^J \|\boldsymbol{x}_j\|_p$

List of Figures

Chapter 1

Introduction

1.1 General Motivation

Most traditional approaches in machine learning focus on learning on one single isolated data set. This holds true for supervised as well as for unsupervised learning methods. It is clear that the restriction to learn on isolated data sets neglects certain fundamental aspects of human learning. Humans approach a new learning task on the basis of knowledge gained from previous learned tasks. Learning would be a lot more difficult if knowledge gained from earlier tasks could not be used to learn a new related task. Thus, transfer of knowledge is an essential element in learning. The process of transferring knowledge over related tasks or views of data is called *transfer learning*. Examples for transfer learning in human life are when one finds it easier to learn the rules of a new card game having already learned another card game or to learn a Romance language like Spanish or French by already being proficient in Italian. This process of transfer learning across tasks that is very natural for humans constitutes a major problem in machine learning. When different tasks are related, it can be advantageous to learn all tasks simultaneously instead of following the more traditional approach of learning each task independently of the others.

In this thesis, we present novel methods for transfer learning, both in supervised and in unsupervised learning problems.

We approach the problem of learning data representations that are common across multiple related tasks in a supervised learning setting. *Multi-task learning* is one way of achieving inductive transfer between different tasks or instances. The principle goal of transfer learning is to improve generalization

performance by using information available across all related tasks. Relatedness of tasks is the key to the multi-task learning approach. Obviously, one cannot expect that information gained through the learning of a set of tasks will be relevant to the learning of another task that has nothing in common with the already learned set of tasks. When the tasks are related, joint learning usually performs better than learning each task independently. Learning jointly over related tasks is of special importance when only few data points are available per task. In such cases, independent learning is not successful. Moreover, learning common *sparse* representations across multiple tasks or data sets may also be of interest as sparse solutions are much easier to interpret. While the problem of learning sparse representations has been extensively studied for single-task supervised learning (e.g., using 1-norm regularization), there has been done only limited work in the multi-task supervised learning setting. In the first part of this thesis we close this gap. We evaluate a class of regularizers which are used for multi-task learning in terms of prediction and interpretability of solutions. The class of regularizers we formally study addresses both problems, coupling of tasks and enforcing sparsity.

The methods we consider in the first part of the thesis need vectorial data as input data. Often, however, no access is given to the underlying vectorial representation of the data, but only pairwise distance are measured, especially in biological and medical problems. Relational data, or distance data, is in no natural way related to the common viewpoint of objects lying in some well behaved space like a vector space. A loss-free embedding of relational data into a vector space is usually not possible.

In the second part of the thesis we approach this problem and develop unsupervised Bayesian clustering methods that are able to work on distance data directly. First, we present a flexible probabilistic clustering method that is rotation- and translation- invariant. A Dirichlet process prior is used to partition the data. In a second step we approach the transfer-learning problem in unsupervised learning: the goal is to learn the common structure across multiple views of co-occurring samples instead of learning on every view separately. Here *multi-view* learning is one way of achieving inductive transfer between different views of a phenomenon. The aim is to use the relationship between these views to improve the learning process and to learn simultaneously from two or more data sets with co-occurring observations.

Despite the strong presence of medical and biological applications it is important to notice that the methods are not restricted to biomedical problems. The methods are very generic and cover a broad field of application.

1.2 Outline and Contributions

After giving a brief overview of the main ideas of this thesis, we now present a more detailed roadmap of how this work is organized in the forthcoming chapters.

This thesis is divided into two parts, the first part addresses supervised data analysis and the second part unsupervised data analysis. Chapter 2 functions as a general introduction to both areas of data analysis. It consists of sections that are on some extent detached from each other but provide necessary background for the thesis.

In the first part of the thesis, we concentrate on the problem of variable selection in supervised learning problems. Chapter 3 lays the foundation for variable selection in linear regression models. The need for sparse learning algorithms is explained. A method for single variable selection, the Lasso, as well as the Group-Lasso for grouped variable selection are introduced. Finally, the multi-task problem setting is presented and the use of the Group-Lasso to solve multi-task learning problems is explained.

In Chapter 4 we present one of the main contributions of this thesis: a complete analysis of the $\ell_{1,p}$ Group-Lasso. We characterize conditions for solutions of the Group-Lasso for *all* p-norm regularizations and we present a highly efficient unified active set algorithm with convergence guarantee. This new method is then tested on many real-world multi-task data sets where the main application area lies in the field of biomedical data analysis.

In the second part of the thesis, we look at unsupervised learning problems. Chapter 5 introduces partition processes and the Gauss-Dirichlet clustering process which constitute the basis for the subsequent analysis. While the first part of the thesis concerned vectorial data, the second part concentrates on a different aspect of data analysis that is of high importance: the focus is set on data that is not available in vectorial form, but solely in form of pairwise distances.

In Chapter 6 we present the second main contribution of the thesis: a probabilistic clustering approach to cluster distance data. This Bayesian clustering method is translation- and rotation- invariant and enables to work on distance data directly. No embeddings into a vector space are needed. A highly efficient sampling algorithm is presented. Finally, we even go beyond learning on single instances and consider the transfer learning problem on distance data. We extend the novel model in a way that it is able to cluster multiple views of co-occurring samples.

In summary, two main types of contributions are presented in this thesis. First, we present a novel theory in the field of supervised multi-task learning. Second, we introduce a novel method in the area of unsupervised learning to cluster distance data which is able to partition data that is either available as single instances or as multiple views. Figure 1.1 illustrates the topics discussed in this thesis.

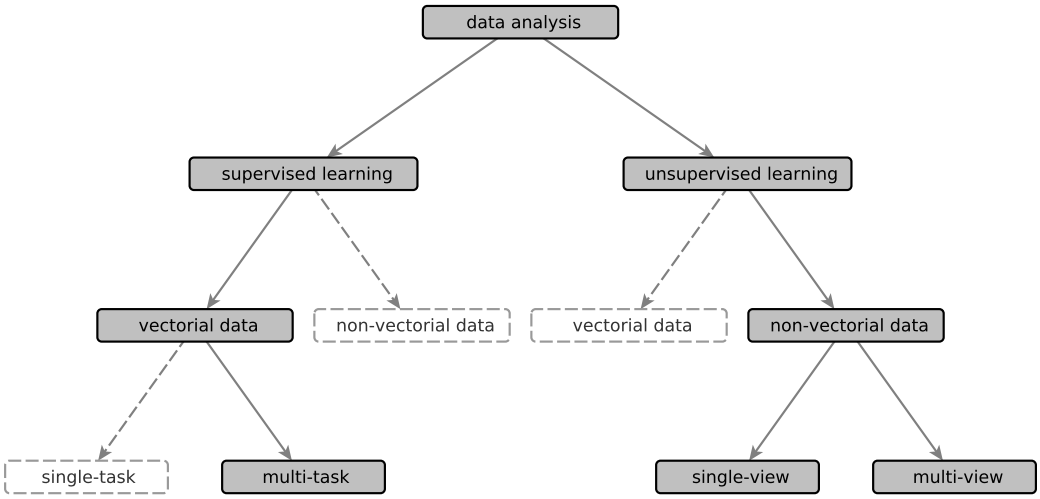

Figure 1.1: Illustration of the organization of the thesis. Discussed topics are highlighted.

The following publications have resulted out of the work presented in this thesis:

- A Complete Analysis of the $\ell_{1,p}$ Group-Lasso.
  Julia E. Vogt and Volker Roth.
  Proceedings of the 27th International Conference on Machine Learning, 2012.

- The Group Lasso: $\ell_{1,\infty}$ Regularization versus $\ell_{1,2}$ Regularization.
  Julia E. Vogt and Volker Roth.
  Pattern Recognition: 32-nd DAGM Symposium, Lecture Notes in Computer Science, 2010.

- The Translation-invariant Wishart-Dirichlet Process for Clustering Distance Data.
  Julia E. Vogt , Sandhya Prabhakaran, Thomas J. Fuchs, Volker Roth.
  Proceedings of the 27th International Conference on Machine Learning, 2010.

- Interferon-Induced Gene Expression is a Stronger Predictor of Treatment Response Than IL28B Genotype in Patients With Hepatitis C.
  Michael T. Dill, Francois H.T. Duong, Julia E. Vogt, Stephanie Bibert, Pierre-Yves Bochud, Luigi Terracciano, Andreas Papassotiropoulos, Volker Roth and Markus H. Heim.
  Gastroenterology, 2011 Mar;140(3):1021-1031.e10.

- The $\ell_{1,p}$ Group-Lasso for Multi-Task Learning.
  Julia E. Vogt and Volker Roth.
  Workshop on Practical Applications of Sparse Modeling: Open Issues and New Directions *Workshop @ Neural Information Processing Systems*, Whistler, Canada, 2010.

Chapter 2

Background

2.1 Supervised Data Analysis

One of the aims in data analysis is to analyze the relationship between measurements and the corresponding responses that belong to each measurement. The measurements X are referred to as *input* data and the responses y as *target* or response variables. Such a type of learning where not only the input data but also the corresponding targets are observed is known as supervised learning. The aim is to learn the "best" mapping f from the input space to the target space to predict the target values for new unknown data, the *test data*, i.e. a function f that generates values y' that are close to the "real" target values y (see e.g. [Bish 09] for more details). If the target labels are discrete, then the learning problem is called *classification* and we want to predict which category or class a new sample belongs to. In case of continuous labels, we are looking at a *regression* problem. In regression, the aim is to find a function that fits the data points best. The inferred function should predict the correct labels for any new test data. This requires the estimated function f to be able to generalize from training data to unknown test data. In standard learning problems, one data set at a time is considered and the learning problem for every data set is solved separately. An example for classification and regression is depicted in Figure 2.1.

Application of Supervised Data Analysis in Medicine. We present a medical example for the case of supervised classification. The input data consists of gene expression values for various genes measured from various patients suffering from hepatitis C. Each patient either responded to a special

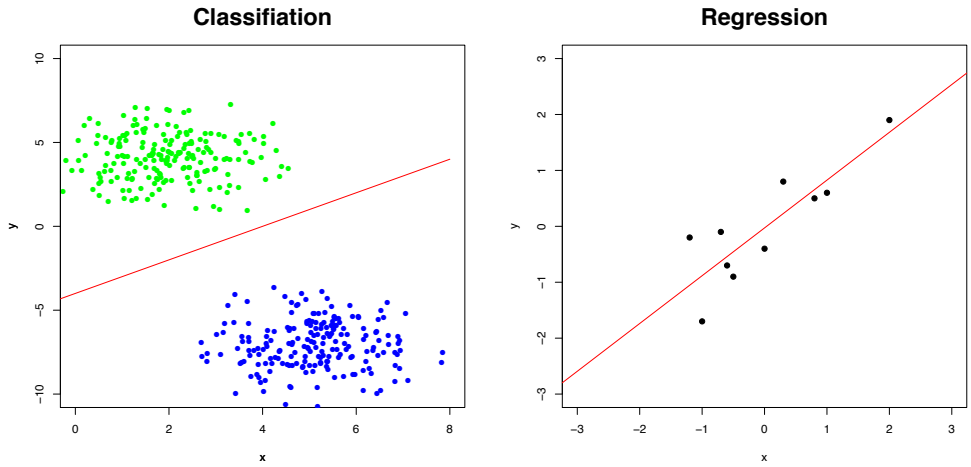

Figure 2.1: A graphical depiction of classification (left) and regression (right). In classification, the aim is to predict which category or class a new sample belongs to whereas in regression one wants to find a function that fits the data points best.

medical treatment or did not respond. The response in this case is a binary variable which can take values $\{0, 1\}$ where 0 and 1 indicate the patients' response and non-response to treatment respectively. The goal is then to learn a function which can take the gene expressions as input and accurately predicts whether a new patient will respond to the treatment or not. Details to this special problem can be found in [Dill 11].

2.1.1 Linear Regression Models

Linear regression models are, due to its simplicity, amongst the most used models for analyzing regression problems. The simplest form of a linear regression model is linear in its input variables and in its parameters and is defined as

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_d x_d \tag{2.1}$$

with input variable $\mathbf{x} = (x_1, \ldots, x_d)^T$, parameters $\beta_0, \ldots, \beta_d$ and corresponding target value y. The goal is to find the optimal regression coefficients β which minimizes the difference between the predicted target value y' and the real target value y.

Ordinary Least Squares (OLS), for instance, is a well-known technique to solve this minimization problem. In case of OLS, the sum of the squared difference between observed and predicted response is minimized to find the optimal values of β. Given a training set with n observations, arranged as the rows of a data matrix $X \in \mathbb{R}^{n \times d}$ and the corresponding target values $\mathbf{y} = (y_1, \ldots, y_n)$, the optimization problem that needs to be solved is a convex one and results in the following:

$$\|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2 \rightarrow \min_{\boldsymbol{\beta}}, \tag{2.2}$$

where $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_d)^T$.

2.1.2 Sparsity in Data Analysis

In the medical classification example mentioned above, one aim was to predict the correct responses for new unseen data. The other prominent goal the medical doctors were interested in was to identify a small subset of genes that are more important in terms of predicting the outcome than the remaining set of variables. By using hundreds or thousands of genes for data analysis, interpretation of the result might be difficult. Moreover, selecting genes in advance is often difficult or might not even be possible. This problem of preselecting genes leads to a different aspect of data analysis: to determine the significance of the input variables in terms of predicting the response. The aim is now to obtain solutions that are easier for the expert to interpret by identifying a small subset of significant variables. Obtaining a small set of genes enables the medical doctors to focus their research efforts on those specific few genes found by the sparse predictor. *Sparse learning* refers to methods of learning that seek a trade-of between prediction accuracy and sparsity of the result. By forcing the solution to be sparse, as in obtaining a sparse set of genes, better interpretability of the model is expected.

2.1.3 Multi-Task Learning

In standard learning problems one data set is considered and the learning problem for this single data set is solved. In case one or more related problems (or tasks) exist, all problems are solved independently of each other. In the following, we look at the non-standard case of learning by simultaneously utilizing the information given by multiple related data sets. *Multi-task* learning is based on the assumption that multiple tasks share some features

or structures. By *tasks* we denote related data sets that share the same set of features but stem from different measurements. The aim is to profit from the amount of information given by all data sets together. This is especially important if every single data set consists of only few data points. Especially in biomedical applications, often high dimensional data is available but sample size is small. This problem arises for instance in gene expression measurements by measuring the expression values of tens of thousands of genes of only a few patients. In multi-task learning the aim is to learn on many related data sets simultaneously and hence be able to get a better prediction than on learning on every of these data sets separately. In terms of variable selection, this means that the problem of joint variable selection across a group of related tasks is considered instead of single variable selection per task. The multi-task scenario is illustrated in Figure 2.2. Experimental work showing the benefits of such transfer learning relative to individual task learning are given, for instance, in [Caru 97], [Oboz 06], [Yu 07], [Argy 07] or [Bick 04].

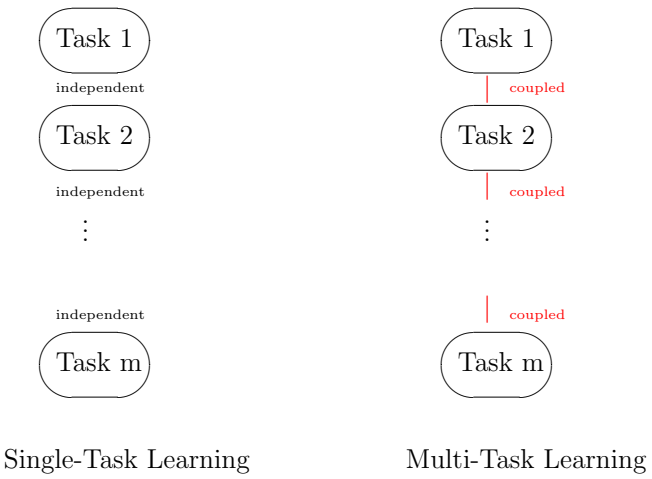

Figure 2.2: Single-task learning versus multi-task learning: In single task learning, all tasks are solved independently of each other whereas in multi-task learning the tasks are coupled. This coupling allows to learn over all data sets simultaneously.

2.2 Convex Optimization

Convex optimization plays an important role in our work on multi-task learning. Here, we briefly remind of the basics of convex optimization. For two convex and continuously differentiable functions f and g the general constrained convex optimization problem reads as:

$$(P) \quad \begin{cases} f(\boldsymbol{x}) \longrightarrow \min \\ g(\boldsymbol{x}) \leq 0 \end{cases}$$

For convex problems some nice properties hold, e.g., every local solution is also a global solution and if f is strictly convex and an optimum exists, then the optimum is unique.

The *Lagrangian function* $\mathcal{L}$ to the problem (P) is defined as a weighted sum of the objective function and the constraint function, i. e.

$$\mathcal{L}(\boldsymbol{x}, y) := f(\boldsymbol{x}) + \lambda g(\boldsymbol{x})$$

for $\lambda \in \mathbb{R}_+$. λ is called the *Lagrangian multiplier* or *Lagrangian dual variable*.

A problem that is closely related to (P) is the so-called Lagrange *dual function* associated with (P). The dual problem is defined by:

$$(D) \quad \begin{cases} \varphi(\lambda) := \inf\limits_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \lambda) \longrightarrow \max \\ \lambda \geq 0 \end{cases}$$

In general, it is not guaranteed that the dual problem has a solution, even if the primal problem has a solution, as well as the other way round.

In convex optimization, if Slater's condition is fulfilled (i. e. if a feasible vector $\tilde{\boldsymbol{x}}$ exist so that $g(\tilde{\boldsymbol{x}}) < 0$), then strong duality holds, i. e. $\inf(P) = \sup(D)$. Strong duality implies that the constrained primal problem (P) and the penalized Lagrangian problem $\mathcal{L}$ are related in the following way: any primal feasible solution $(\tilde{\boldsymbol{x}}, \tilde{\lambda})$ of (D) is also a solution to (P). On the other hand, if an optimum $\tilde{\boldsymbol{x}}$ to (P) exists, then there also exists a λ so that $(\tilde{\boldsymbol{x}}, \lambda)$ optimize (D). This observation is extremely useful, especially in cases when the dual problem is easier to solve than the primal problem.

2.3 Bayesian Inference

In the second part of the thesis, a Bayesian clustering model is presented. In this Section we explain the basics of a probabilistic Bayesian view point of an optimization problem. The first component of Bayesian analysis consists of a prior belief over the parameters $\boldsymbol{\theta}$ of a model before any data is observed which might change this prior belief. This prior belief is represented in the form of a probability distribution $p(\boldsymbol{\theta})$. The second component of Bayesian analysis is the likelihood function. The observations, denoted by D, are modeled by the likelihood function $p(D|\boldsymbol{\theta})$, which quantifies how well the parameters explain the observed data. The goal is to model the effect of the observations on the prior belief over $\boldsymbol{\theta}$. Such an effect can be obtained using Bayes theorem:

$$\begin{aligned} p(\boldsymbol{\theta}|D) &= \frac{p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(D)} \\ &\propto p(D|\boldsymbol{\theta})p(\boldsymbol{\theta}) \end{aligned} \tag{2.3}$$

$p(D)$ denotes the normalization constant. Using Bayes' theorem, we obtain $p(\boldsymbol{\theta}|D)$, the so-called *posterior* distribution over $\boldsymbol{\theta}$. The posterior distribution models the posterior belief in $\boldsymbol{\theta}$ based on observed data. The optimal value of $\boldsymbol{\theta}$ can be found by maximizing the posterior distribution over $\boldsymbol{\theta}$.

2.4 Unsupervised Data Analysis

Unsupervised data analysis refers to learning problems where the training data consists of a set of input vectors without any corresponding target values like in supervised learning. Unsupervised partitioning or clustering aims at extracting hidden structure from data. Figure 2.3 illustrates the processes of supervised and unsupervised learning. An important research area in unsupervised learning is probabilistic modeling. Here the underlying assumption is that a generative model exists that captures the hidden structure of the data. In unsupervised clustering, one finds such a probability distribution that models this hidden structure. We briefly introduce the well known finite and infinite mixture models in the next Section. We also illustrate with an application example how these clustering models were used in recent medical research concerning the treatment of chronic hepatitis C. In Chapter 6, we extend these well known clustering concepts from vectorial to distance data and we show how the hidden structure of the data can be learned not only on single instances but even on multiple data sets.

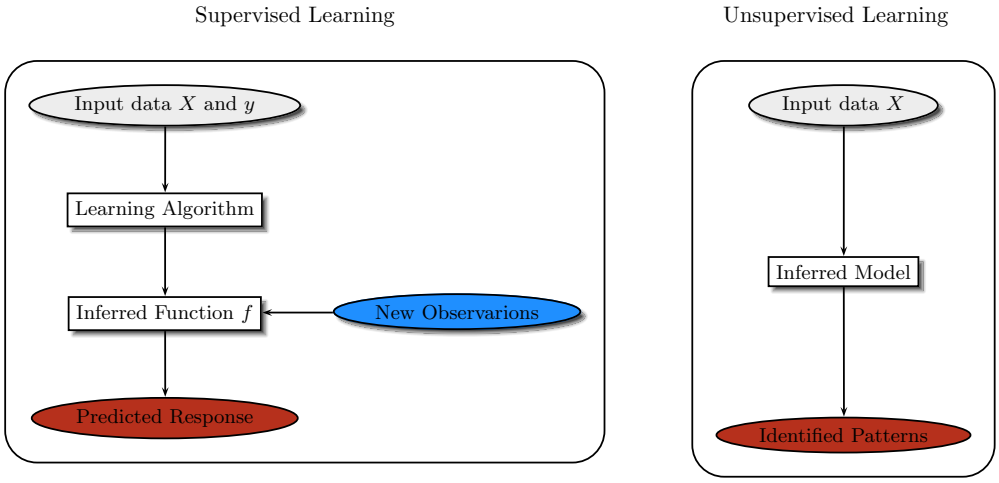

Figure 2.3: A graphical depiction of the supervised and unsupervised learning problem. The supervised learning problem (left) involves the learning of a relationship between input and response variables. The observations are used to train a learning algorithm which is then used for predicting responses for new inputs. In unsupervised learning, no target values are given. The aim is to find a model which extracts patterns within the given data.

2.4.1 Finite and Infinite Mixture Models for Clustering

In this Section we introduce the well-known concept of mixture models for clustering. A cluster denotes a group of similar data points. From a Bayesian perspective, one cluster can be interpreted as one component of a mixture model, and the data points which belong to this cluster are assumed to be sampled from the same distribution. Learning the underlying clustering structure basically means learning the parameters for each component distribution of the mixture model. The assumption is that every object, i.e. every data point, belongs to one class b and that the assignment of an object x to a class is independent of the assignments of all other objects. For K classes, i.e. for K clusters and for n d-dimensional observations arranged in a data matrix $X \in \mathbb{R}^{n \times d}$ the probability of all n objects reads as

$$p(X|\boldsymbol{\theta}) = \prod_{i=1}^{n} \sum_{k=1}^{K} p(\boldsymbol{x}_i|b_i = k)\theta_k. \tag{2.4}$$

where θ_k denotes the weight of class k and $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$ a variable with a prior distribution $p(\boldsymbol{\theta})$.

By using a Bayesian mixture model, every data point belongs to one cluster with a special probability, the output of the clustering is a probability distribution. There exist two types of clustering frameworks, the finite and the infinite mixture models. *Finite* mixture modeling means that there are a fixed number K of components in the mixture. This corresponds to learning a fixed number of clusters for the given data in contrast to infinite mixture models. The mixture of Gaussians is a well studied example of a finite mixture model, see e.g. [Cord 01]. Popular clustering algorithms like k-means are special cases of this method. An important question in finite mixture modeling is how the number of mixture components K is chosen. One way to handle this question is to use cross-validation, a standard technique for model selection. However, cross validation often leads to high computational costs because the model needs to be trained many times with different values for K. Then, K with the highest likelihood on some held-out data is chosen. This problem can be circumvented by using an *infinite* mixture model, where in principle infinitely many clusters are feasible. The extension from finite to infinite mixture models leads to a Dirichlet process mixture model, formally discussed in [Ferg 73]. The Dirichlet process denotes a nonparametric Bayesian framework for mixture models. In the case of infinitely many classes, equation (2.4) changes to the following for $K \to \infty$:

$$p(X|\boldsymbol{\theta}) = \prod_{i=1}^{n} \sum_{k=1}^{\infty} p(\boldsymbol{x}_i|b_i = k)\theta_k \tag{2.5}$$

Specifically, a distribution on partitions of objects is defined and the probability of a partition is independent of the ordering of the objects. We will briefly explain a process that induces a distribution on partitions, the well-known *Chinese Restaurant process*, see for instance [Ewen 72, Neal 00, Blei 06].

The Chinese Restaurant Process

The Chinese restaurant process was introduced by Jim Pitman [Pitm 06] and relies on the following metaphor: imagine a Chinese restaurant with countably infinitely many tables. Objects that are supposed to get clustered correspond to customers and the clusters correspond to tables at which the customers sit. Customers walk in, one after another and sit down at some of the tables. A customer chooses a table according to the following random process:

1. The first customer always chooses the first table.

2. The n-th customer chooses the first unoccupied table with probability $\frac{\alpha}{n-1+\alpha}$ where α is a scalar parameter and an already occupied table with probability $\frac{c}{n-1+\alpha}$, where c denotes the number of people sitting at that table.

This process continues until all customers are seated and defines a distribution over the allocation of customers to tables. Any seating arrangement creates a partition. Thereby, the probability of a seating is invariant under permutations, it is an *exchangeable* partition process. The Chinese restaurant process is an intuitive example that demonstrates how a prior for an infinite mixture model can be specified and it shows a sequential process that generates exchangeable cluster assignments. This process is illustrated in Figure 2.4 on an example of clustering genes.

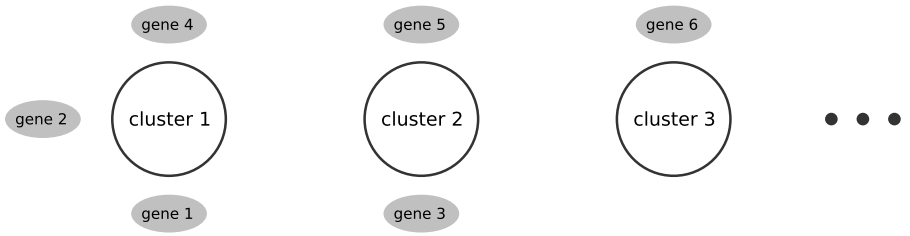

Figure 2.4: Illustration of the Chinese restaurant process, with genes corresponding to customers and clusters corresponding to chosen tables.

Application of Clustering in Medicine. As an example for clustering, we again present a medical example. In this study, the pharmacodynamics of a drug at various time points during the treatment of chronic hepatitis C were investigated. We used a Bayesian infinite mixture model as explained in Section 2.4.1 to cluster the gene expression data that was obtained from liver biopsies. By using a Dirichlet process prior, we did not need to fix the number of clusters in advance. Based on their expression values over time, three distinct gene clusters were identified which correlated with tumor differentiation grade, tumor size and enrichment of specific signaling pathways. This work was presented at the International Liver Congress (ILC) 2012 in Barcelona and a manuscript is in progress.

2.4.2 Multi-View Learning

In this Section we consider the problem of clustering multiple instances in parallel instead of single instances independently of each other. The idea is to get a better representation by jointly using multiple views of the same underlying phenomenon and improve performance of the learning algorithm. Clustering data that is available in multiple instances is a problem in the area of transfer learning. The aim is to learn from two or more data sets with co-occurring observations and to use all the available information instead of formulating separate problems. Increased performance compared to traditional single-view learning has been reported in various applications (see e.g. [Chau 09], [Bick 04] or [Bick 05]). Assume there are two random vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2 \in \mathbb{R}^d$ that each characterize the same object, but in different views. Both vectors are Gaussian distributed $\boldsymbol{x}_1|z \sim \mathcal{N}(\boldsymbol{\mu}^z_{x_1}, \Gamma_{x_1})$ and $\boldsymbol{x}_2|z \sim \mathcal{N}(\boldsymbol{\mu}^z_{x_2}, \Gamma_{x_2})$, where $\boldsymbol{\mu}^z_{x_1}$ and $\boldsymbol{\mu}^z_{x_2}$ denote the mean vector in view 1 and view 2 corresponding to cluster z. The model (cf. [Klam 06]) then reads:

$$z \sim \text{Mult}(\theta) \tag{2.6}$$

$$(\boldsymbol{x}_1, \boldsymbol{x}_2)|z \sim \mathcal{N}(\boldsymbol{\mu}^z, \Gamma), \tag{2.7}$$

which corresponds to a standard mixture of Gaussians. $\boldsymbol{\mu}^z$ and Γ denote the joint mean vector and covariance matrix. Using a full covariance matrix

$$\Gamma = \begin{pmatrix} \Gamma_{x_1} & \Gamma_{x_1 x_2} \\ \Gamma_{x_2 x_1} & \Gamma_{x_2} \end{pmatrix} \tag{2.8}$$

leads to a model that does not differentiate between dimensions and views. This coincides to single-view clustering in the augmented space, also called *product space*.

A special case of the multi-view setting is the so-called *dependency-seeking* clustering [Klam 06]. Here, the underlying assumption is that views are conditionally independent of each other given some cluster structure. Such a model is supposed to identify dependencies between data sets and thus reveals shared information. The idea behind dependency-seeking clustering is to find a coherent structure among all views that is based on their interdependencies. This is achieved by replacing the previous covariance matrix (2.8) with the following:

$$\Gamma = \begin{pmatrix} \Gamma_{x_1} & 0 \\ 0 & \Gamma_{x_2} \end{pmatrix} \tag{2.9}$$

The dependency-seeking aspect is caused by the off-diagonal zero-entries which effectively forces the model to uncover between-view dependencies based on a common cluster structure.

2.5 Summary

In this chapter we introduced some basic concepts of both supervised and unsupervised learning problems that we will use throughout the following chapters. The next chapter concentrates on supervised learning and functions as an introduction to variable selection in linear regression models. The concepts and ideas we present in Chapter 3 constitute the foundation for our work on multi-task learning in Chapter 4.

Chapter 3

Variable Selection in Linear Regression Models

3.1 Introduction to Linear Regression Models

We already mentioned briefly in Chapter 2 how data analysis is accomplished with the use of regression models. In this section we explain the general setup of linear regression models. Given a d-dimensional input variable $\boldsymbol{x} \in \mathbb{R}^d$ the goal in linear regression models is to predict a corresponding real-valued response variable $y \in \mathbb{R}$. The relationship between these two variables is defined based on a function which is *linear* in the regression coefficients $\boldsymbol{\beta} = (\beta_1, \beta_2, ..., \beta_d)^T$ and possibly nonlinear in its basis functions $\boldsymbol{\phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), ..., \phi_d(\boldsymbol{x}))^T$ and reads as $y = \boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{\beta}$.

If we obtain a set of independent and identically distributed (i.i.d) observations $D = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$, our goal is to find the value of $\boldsymbol{\beta}$ which best explains the observations. The estimation of the optimal $\boldsymbol{\beta}$ is done by defining a likelihood function $l(\boldsymbol{\beta})$ which quantifies how well the data is explained based on the given parameter $\boldsymbol{\beta}$. The goal of inference is to find the parameter $\boldsymbol{\beta}$ which maximizes the likelihood function. This results in the following maximization problem

$$l(\boldsymbol{\beta}) \rightarrow \max_{\boldsymbol{\beta}}. \tag{3.1}$$

Often, the equivalent minimization problem

$$-\ln(l(\boldsymbol{\beta})) \rightarrow \min_{\boldsymbol{\beta}} \tag{3.2}$$

is considered instead, where "ln" denotes the natural logarithm function: the logarithm function is monotonically increasing, hence maximizing a function is equivalent to maximizing its log or minimizing the negative log.

Hence equation (3.1) and equation (3.2) represent two views of the same optimization problem. In the following, the objective function $-\ln(l(\boldsymbol{\beta}))$ is referred to as cost function.

By inferring an optimal function the aim is to be able to generalize the relationship between given input and response for unknown data. The objective is to be able to predict responses for new inputs where the true targets are unknown. This notion of *generalization* can be quantified by measuring the error, called *prediction error*, made in predicting responses for unseen data. The lower the error, the better is the generalization capacity of the model.

3.2 Generalized Linear Models

So far we discussed linear regression models where the response variables consisted of real-valued scalars. To be able to handle other types of response variables like binary values or count data, we now introduce an extension of the concept of linear models, the generalized linear model (GLM). According to [McCu 83], a generalized linear model consists of three elements:

1. The first element is a *random* component $f(y;\mu)$ specifying the stochastic behavior of a response variable y which is distributed according to some distribution with mean μ.

2. The second part of the model consists of a *systematic* component of the model. It is a description of the vector $\eta = \boldsymbol{x}^T\boldsymbol{\beta}$, specifying the variation in the response variable accounted for by known covariates $\boldsymbol{x}$ for some unknown parameters $\boldsymbol{\beta}$.

3. The third component is described by a *link* between the random and the systematic part of the model. The link function $\nu(\mu) = \eta$ specifies the relationship between the random and systematic components.

Classical linear models employ a normal distribution in the random component and the identity function as link function.

GLMs allow us to replace the normal likelihood by any exponential family distribution as random component and to use any monotonic differentiable function ν as link function.

A distribution from the exponential family has the following form:

$$f(y;\theta,\phi) = \exp(\phi^{-1}(y\theta - b(\theta)) + c(y,\phi)), \tag{3.3}$$

with natural parameter θ, sufficient statistics y/ϕ, log partition function $b(\theta)/\phi$ and a scale parameter $\phi > 0$.

In model (3.3), the mean of the responses $\mu = E_\theta[y]$ is related to the natural parameter θ by $\mu = b'(\theta)$. The link function ν can be any strictly monotone differentiable function. In the following, however, we will consider only *canonical* link functions for which $\nu(\mu) = \eta = \theta$. We will thus use the parametrization $f(y;\eta,\phi)$.

From a technical perspective, an important property of this framework is that $\log f(y;\eta,\phi)$ is strictly concave in η. The concavity follows from the fact that the one-dimensional sufficient statistics y/ϕ is necessarily *minimal*, which implies that the log partition function $b(\eta)/\phi$ is strictly convex, see [Brow 86, Wain 05].

The standard linear regression model is a special case derived from the normal distribution with $\phi = \sigma^2$, the identity link $\eta = \mu$ and $\nu(\eta) = (1/2)\eta^2$. Other popular models include logistic regression (binomial distribution), Poisson regression for count data and gamma-models for cost- or survival analysis.

3.3 Regularization in Linear Models

In case of OLS, as introduced in Section 2.1, the problem of minimizing the cost function for some data matrix X, labels $\boldsymbol{y}$ and coefficients $\boldsymbol{\beta}$ is the following:

$$\|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2 \to \min_{\boldsymbol{\beta}} \tag{3.4}$$

However, often OLS performs poorly, due to *over-fitting*. This phenomenon can arise if the space of possible functions over which the optimization is done, the so-called *hypothesis space*, allows a very rich set of functions. It might happen that the resulting optimal function fits the training data perfectly, but performs poorly in prediction because the estimation is tuned specifically for the training data. The reverse problem can happen as well: if the hypothesis space is chosen to be too restrictive, under-fitting can occur due to the restrictive choice of possible functions. By introducing some *regularization* term, the over- and under-fitting phenomenon can be controlled

and the OLS solution can be improved. A well known regularization technique called ridge regression consists in adding a penalty term to penalize large $\boldsymbol{\beta}$-values:

$$\|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2 \rightarrow \min_{\boldsymbol{\beta}} \tag{3.5}$$

λ denotes a Lagrangian parameter that governs the importance of the regularization. Equivalently we can look at the constrained optimization problem by adding a feasible region to the optimization problem (3.4) and forcing the coefficients to lie within that feasible region:

$$\|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2 \rightarrow \min_{\boldsymbol{\beta}} \tag{3.6}$$

$$s.t. \quad \|\boldsymbol{\beta}\|_2^2 \leq \kappa \tag{3.7}$$

κ denotes a parameter that defines the size of the feasible set. By adding this constraint, the coefficients are restricted to small values.

Techniques of this kind that reduce the value of the coefficients are called *shrinkage* methods. As mentioned in Section 2.1.2, often it is not only desired to obtain a low prediction error but also a model that is easy to interpret. While dealing with large sets of predictor variables, usually it is desirable to select a small set of significant variables which have a strong effect on the response variable. The selection of a small subset of variables is especially important from an application point of view, e.g. when dealing with gene expression data sets. Often, the expert needs to know which genes are the most important or significant ones for prediction out of a set of tens of thousands of genes. This process of selecting significant variables is called *feature selection*. Feature selection can be interpreted as estimating a set of regression coefficients for the significant predictor variables which results in a *sparse* vector $\boldsymbol{\beta}$. A variable x_i in this interpretation is called significant if the corresponding value $\beta_i \neq 0$. Ridge regression is an effective tool to control the phenomenon of over-fitting by shrinking the coefficients, but it is not sufficient for variable selection as it does not force the solution to be sparse. To separate out the more significant variables from lesser significant ones requires an extra selection step after obtaining the $\boldsymbol{\beta}$ estimates. This filtering of significant variables can be obtained by using sparse regularization techniques as will be discussed in the following.

3.4 Single Variable Selection - The Lasso

A promising technique of sparse regularization called the *Lasso* was proposed by Tibshirani in [Tibs 96]. The Lasso regularization consists in adding an ℓ_1-norm regularization to the cost function, as opposed to the ℓ_2-norm regularization. This regularization has the effect of shrinking the $\boldsymbol{\beta}$ parameters as in ridge regression, but in addition it forces the solution to be sparse:

$$\|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 \rightarrow \min_{\boldsymbol{\beta}} \tag{3.8}$$

The constrained form of this problem is the following:

$$\|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2 \rightarrow \min_{\boldsymbol{\beta}} \tag{3.9}$$

$$s.t. \qquad \|\boldsymbol{\beta}\|_1 \leq \kappa \tag{3.10}$$

The main advantage of the Lasso is that it does both, continuous shrinkage and automatic sparse variable selection. Figure 3.1 shows the least squares cost function and the constraint region for ridge regression and the Lasso. It illustrates that the variables selected by ridge regression are shrunk, but not sparse due to the spheric form of the feasible set and the variables selected by the Lasso are shrunk and in addition encouraged to be sparse by using the ℓ_1-norm constraint.

By using different values for the model parameter κ (or λ in the Lagrangian version), different models are obtained. Hence κ can be viewed as a model selection parameter which also has to be inferred as a part of the learning process which is usually done via cross-validation. In cross-validation, the training data is divided into two parts, a training set to train the model with a fixed value of κ and a test set. The test set is used for calculating the prediction performance of the model. For each κ, this procedure is averaged out for different divisions of the training set on a range of values for κ. The value of κ that yields the best accuracy is chosen and the full training data is then used to obtain the final Lasso estimates.

3.5 Grouped Variable Selection - The Group-Lasso

The Lasso was extended by Turlach et. al. ([Turl 05]) and by Yuan and Lin ([Yuan 06]) to the problem, where explanatory factors are represented as

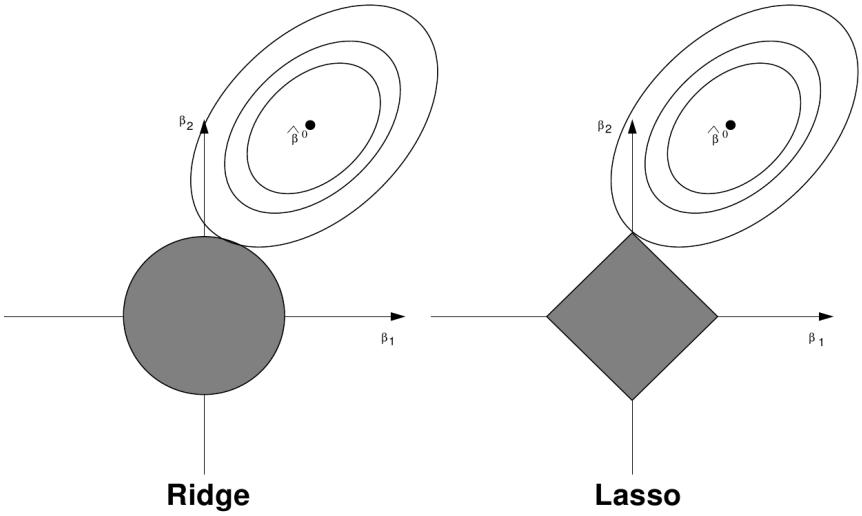

Figure 3.1: Left panel: regularization via ridge regression. Right panel: regularization with the Lasso

groups of variables, leading to solutions that are sparse on the group level. This method that selects sparse groups instead of single variables is called *Group-Lasso*. The Group-Lasso penalty is defined as the sum over the norm of groups of covariates.

More specifically, the $\ell_{1,2}$ Group-Lasso problem for OLS for J groups is the following

$$\|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2 + \lambda(\sum_{j=1}^{J} \|\boldsymbol{\beta}_j\|_2) \rightarrow \min_{\beta} \tag{3.11}$$

or, in constrained form:

$$\|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2 \rightarrow \min_{\beta} \tag{3.12}$$

$$s.t. \quad \sum_{j=1}^{J} \|\boldsymbol{\beta}_j\|_2 \leq \kappa \tag{3.13}$$

$\boldsymbol{\beta}_j$ denotes a sub-vector of $\boldsymbol{\beta}$ which represents all regression coefficients of group j. The constraint consists in the sum over the ℓ_2-norm, which is called

$\ell_{1,2}$-norm. The general constraint $\sum_{j=1}^{J} \|\boldsymbol{\beta}_j\|_p$ is referred to as $\ell_{1,p}$-norm. In principal, any $\ell_{1,p}$-norm can be used for regularization in (3.13) or (3.11).

While a lot of emphasis has been put on analyzing the $\ell_{1,2}$- and $\ell_{1,\infty}$-norms, it remains unclear which general $\ell_{1,p}$-norm is to be preferred under which conditions. A formal characterization of the solution for general $\ell_{1,p}$-norms is missing, and practical comparison experiments are difficult due to the lack of efficient algorithms for any $p \notin \{2, \infty\}$. One main contribution of this thesis is to overcome these problems by providing a formal characterization of the solution and by developing efficient algorithms for all $\ell_{1,p}$-norms.

In Figure 3.2, the OLS cost function and the $\ell_{1,2}$ Group-Lasso constraint region are illustrated in three dimensions for two groups. The feasible set is a cone and the optimum of the function is most likely found on the tip of the cone where one group is set to zero which leads to sparsity on the group-level. Figure 3.3 shows the same scenario with the $\ell_{1,\infty}$ Group-Lasso constraint.

3.6 The Group-Lasso for Multi-Task Learning

As mentioned in Section 2.1.3, if many related tasks are available, prediction can be optimized by learning over all tasks simultaneously instead of handling single tasks separately. One possibility for dealing with multi-task problems is the Group-Lasso we introduced in Section 3.5. The motivation for using the Group-Lasso is to couple the individual tasks via the group-structure in the constraint term.

We will explain the multi-task problem setting as we consider it in this work on the example of the MovieLens data set.[1] MovieLens contains 100,000 ratings for 1682 movies from 943 users. Every user ranks some movies in a five-point scale (1, 2, 3, 4, 5). The genre information of the movies is used as features. Every user defines a task, hence we have 943 tasks and 19 features in this data set, as the information about 19 movie genres is available. Figure 3.4 illustrates the MovieLens data set.

The aim now is to predict how a new movie would be ranked. In standard learning, this learning problem would be solved for every user separately. The problem is that every single user only ranks a small number of movies, hence sample size per user is small. Usually, this leads to poor prediction

[1] The data is available at http://www.grouplens.org.

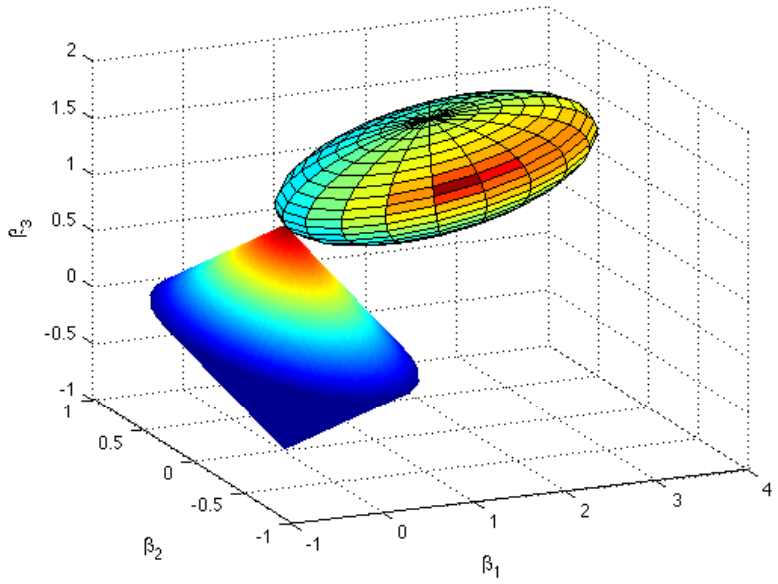

Figure 3.2: $\ell_{1,2}$ Group-Lasso with constraint $\sum_{j=1}^{2} ||\boldsymbol{\beta}_j||_2 \leq 1$: feasible set is a cone.

accuracy. However, it seems to be reasonable to assume that all users share some preferences in ranking movies. This explains the great success of some movies and the flop of others. One alternative to learning on every task separately could be to simply pool the data to one big data set and to use the information jointly given by *all* users. Pooling the data basically means assuming that there was one single user that ranked all movies. The problem with this approach is that although we assume that the users are similar in a way, they are not exactly the same. They differ in age, gender and movie preferences. Hence, just pooling the data is not a good idea either. The approach which seems to be most promising is to couple the different tasks and learn over all tasks simultaneously. This is exactly the multi-task approach we will explain in more detail in the following.

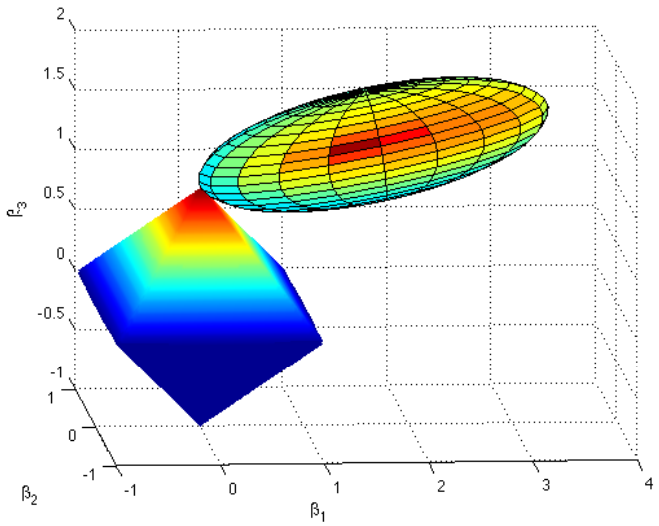

Figure 3.3: $\ell_{1,\infty}$ Group-Lasso with constraint $\sum_{j=1}^{2} ||\boldsymbol{\beta}_j||_\infty \leq 1$: feasible set is a pyramid.

In general, in multi-task learning we obtain multiple tasks. In the following we illustrate how the data looks like for m tasks, d features and total sample size n, where n is split up into sample size n_i for every task i, i.e. $n = \sum_{i=1}^{m} n_i$. The data matrix $X_i \in \mathbb{R}^{n_i \times d}$ for task i has the following form

$$X_i = \begin{array}{c} \begin{array}{cccc} \text{feature 1} & \text{feature 2} & \cdots & \text{feature d} \end{array} \\ \begin{pmatrix} x_{11}^i & x_{12}^i & \cdots & x_{1d}^i \\ \vdots & \vdots & \cdots & \vdots \\ x_{n_i 1}^i & x_{n_i 2}^i & \cdots & x_{n_i d}^i \end{pmatrix} \end{array} := \begin{pmatrix} \boldsymbol{x}_1^i & \boldsymbol{x}_2^i & \cdots & \boldsymbol{x}_d^i \end{pmatrix}$$

with corresponding target $\boldsymbol{y}^i = \begin{pmatrix} y_1^i \\ \vdots \\ y_{n_i}^i \end{pmatrix}$ and coefficient $\boldsymbol{\beta}^i = \begin{pmatrix} \beta_1^i \\ \vdots \\ \beta_d^i \end{pmatrix}$

In this setting, every feature defines a group, i.e. we consider m tasks and d groups. The multi-task data matrix $X_{MT} \in \mathbb{R}^{n \times dm}$ that will be considered to solve the multi-task learning problem for all m tasks simultaneously by

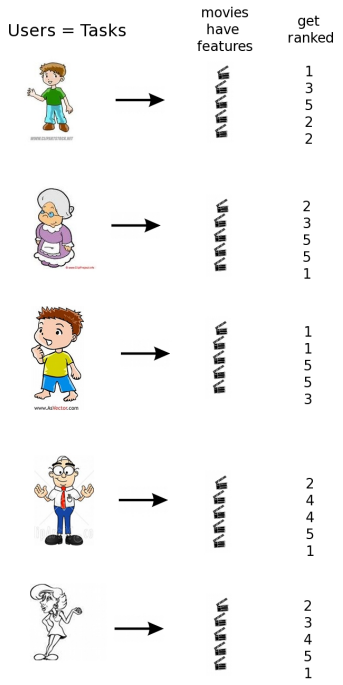

Figure 3.4: Illustration of the MovieLens data set.

coupling the tasks via the group-lasso has the following form

$$X_{\text{MT}} = \begin{pmatrix} \boldsymbol{x}_1^1 & \mathbf{0}_{n_1} & \cdots & \mathbf{0}_{n_1} & & \boldsymbol{x}_d^1 & \mathbf{0}_{n_1} & \cdots & \mathbf{0}_{n_1} \\ \mathbf{0}_{n_2} & \boldsymbol{x}_1^2 & \mathbf{0}_{n_2} & \cdots & \cdots & \mathbf{0}_{n_2} & \boldsymbol{x}_d^2 & \mathbf{0}_{n_2} & \cdots \\ & \ddots & & & & & \ddots & & \\ \mathbf{0}_{n_m} & \mathbf{0}_{n_m} & \cdots & \boldsymbol{x}_1^m & & \mathbf{0}_{n_m} & \mathbf{0}_{n_m} & \cdots & \boldsymbol{x}_d^m \end{pmatrix}$$

where $\mathbf{0}_{n_i}$ denotes a vector of zeroes of length n_i. The corresponding response vector $\boldsymbol{y}_{MT} \in \mathbb{R}^n$ and the coefficients $\boldsymbol{\beta}_{MT} \in \mathbb{R}^{dm}$ have the following form:

$$\boldsymbol{y}_{MT} = \begin{pmatrix} \boldsymbol{y}^1 \\ \vdots \\ \boldsymbol{y}^m \end{pmatrix}, \qquad \boldsymbol{\beta}_{MT} = \begin{pmatrix} \beta_1^1 \\ \vdots \\ \beta_1^m \\ \vdots \\ \beta_d^1 \\ \vdots \\ \beta_d^m \end{pmatrix} := \begin{pmatrix} \boldsymbol{\beta}_1 \\ \vdots \\ \boldsymbol{\beta}_d \end{pmatrix}$$

Hence, the $\ell_{1,p}$ Group-Lasso problem for the OLS likelihood for multi-task learning reads as the following:

$$\|\boldsymbol{y}_{\mathrm{MT}} - X_{\mathrm{MT}}\boldsymbol{\beta}_{\mathrm{MT}}\|_2^2 \to \min_{\beta} \tag{3.14}$$

$$s.t. \quad \sum_{j=1}^{d} \|\boldsymbol{\beta}_j\|_p \leq \kappa \tag{3.15}$$

The motivation for using the Group-Lasso here is to couple the individual tasks via the group-structure of the constraint term. By coupling the tasks we profit from the amount of data given by all tasks together without pooling the data or without being restricted to handling every task separately and suffering from low sample size. Thereby the *coupling strength* between the tasks heavily depends on the choice of the p-norm.

3.6.1 Coupling Strength of ℓ_p-Norms

The coupling properties of the different p-norms have a major influence on the prediction performance of the Group-Lasso variants. The higher the value of p, the stronger the different tasks will be coupled. For $p = 1$, the tasks within one group are barely coupled, as the $\ell_{1,1}$ regularization only induces a global coupling over all tasks. For $p = 2$ there exists an intermediate coupling of tasks within a group and for $p = \infty$ the coupling of the tasks is very strong. This is due to the fact that the ℓ_∞-norm only penalizes the maximum absolute entry of a group, meaning we can increase all other parameters in this group to the maximum value without changing the constraint. Hence we can assign maximum weight to every task in this group. The relation between coupling strength and value of p is illustrated in Figure 3.5 and Figure 3.6. Figure 3.7 shows a graphical representation of different ℓ_p-norms.

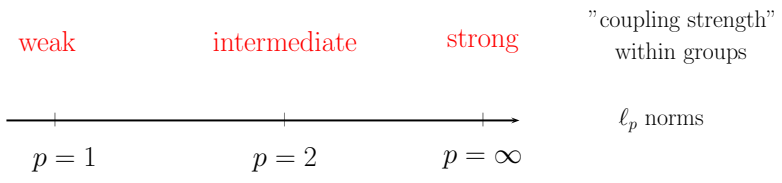

Figure 3.5: Coupling strength of ℓ_p-norms

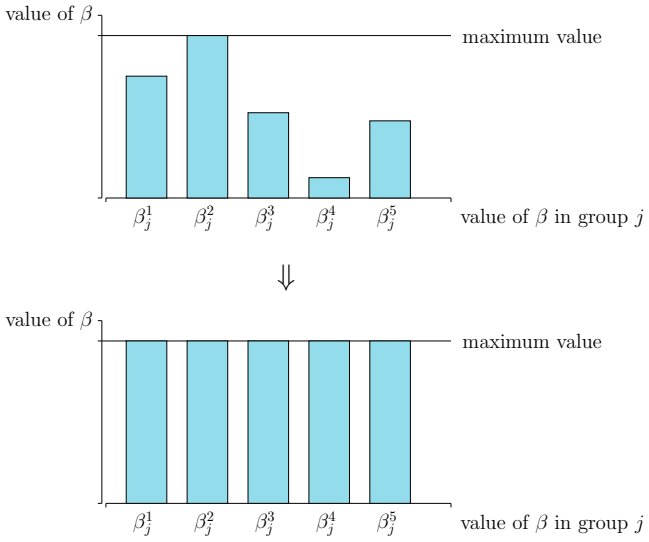

Figure 3.6: For the $\ell_{1,\infty}$ Group-Lasso, all $\boldsymbol{\beta}_j$ in one group can be raised to the maximum value without changing the value of the constraint. This explains the strong coupling properties for $p = \infty$.

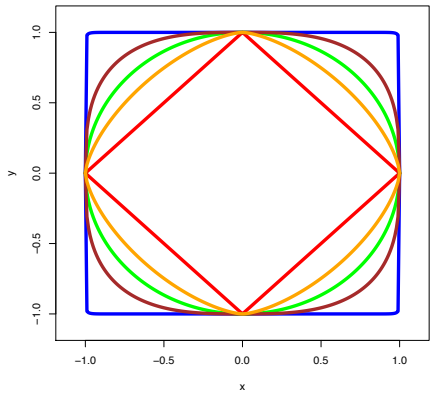

Figure 3.7: Different ℓ_p balls in 2 dimensions: red curve: ℓ_1, orange curve: $\ell_{1.5}$, green curve: ℓ_2, brown curve: ℓ_3, blue curve: ℓ_∞.

3.7 Summary

In this chapter, we introduced variable selection in linear regression models. The need for regularization and, especially, for sparse regularization was explained. The well known technique for sparse variable selection, the Lasso, was presented as well as the Group-Lasso, a method for variable selection on the level of groups of variables. The regularization term for the Group-Lasso differs for varying choices of a p-norm. In this chapter, we explained the use of different $\ell_{1,p}$ regularizers in a multi-task learning scenario where the aim is to couple different, but related tasks over the group-structure of the constraint term. The strength of the coupling heavily depends on the choice of the p-norm. In the next chapter, we will have a close look at the class of $\ell_{1,p}$ regularizers for $1 \leq p \leq \infty$.

Chapter 4

A Complete Analysis of the Group-Lasso

In recent years, mainly two variants of the Group-Lasso have been proposed: one uses the $\ell_{1,2}$-norm and the other one the $\ell_{1,\infty}$-norm as regularization. The $\ell_{1,2}$-norm penalizes the sum of the group-wise ℓ_2-norms of the regression weight, whereas the $\ell_{1,\infty}$-norm penalizes the sum of maximum absolute values per group. Both regularizer induce sparsity on the group level. For $\ell_{1,2}$-constrained problems, extensive research was done, for example in [Yuan 06], [Meie 08], [Argy 07] or [Kim 06]. The solution was characterized by analyzing the optimality conditions by way of subgradient calculus, and conditions for the uniqueness of the solution were formulated. There exist efficient algorithms that can handle large scale problems with input dimension in the millions, see for instance [Roth 08].

Algorithms for the second variant of the Group-Lasso utilizing the $\ell_{1,\infty}$-norm were studied in [Turl 05, Schm 08, Quat 09]. However, questions about the uniqueness of solutions were not addressed in detail, and the method still suffers from high computational costs. Existing algorithms can handle input dimensions up to thousands [Quat 09] or even up to several thousands [Liu 09], but in practical applications these limits are easily exceeded.

The mixed-norm regularization for the $\ell_{1,p}$ Group-Lasso with $1 \leq p < \infty$ was elaborated recently in [Liu 10a] and [Zhan 10], but conditions for the uniqueness of the solution were not formulated so far and for $p \notin \{2, \infty\}$, the available algorithms suffer from high computational costs. For large-scale problems with thousands of groups, existing methods are not efficient. So far, no unified characterization of the solutions for *all* $\ell_{1,p}$ constraints with $1 \leq p \leq \infty$ exists.

In general, the $\ell_{1,p}$ Group-Lasso estimator with $1 \leq p \leq \infty$ has several drawbacks both on the theoretic and on the algorithmic side: (i) in high-dimensional spaces, the solutions may not be unique. The potential existence of several solutions that involve different variables seriously hampers the interpretability of "identified" explanatory factors; (ii) existing algorithms can handle input dimensions up to thousands [Kim 06] or even several thousands [Meie 08], but in practical applications with high-order interactions or polynomial expansions these limits are easily exceeded. For these reasons, large-scale comparisons between the different Group-Lasso variants were computationally intractable; (iii) contrary to the standard Lasso, the solution path (i.e. the evolution of the individual group norms as a function of the constraint) is not piecewise linear, which precludes the application of efficient optimization methods like *least angle regression* (LARS) [Efro 04].

In this chapter we address all these issues: (i) we derive conditions for the *completeness* and *uniqueness* of *all* $\ell_{1,p}$ Group-Lasso estimates, where a solution is called *complete* if it includes all groups that might be relevant in other solutions. This means that we cannot have "overlooked" relevant groups. Based on these conditions we develop an easily implementable *test procedure*. If a solution is not complete, this procedure *identifies all other groups* that may be included in alternative solutions with identical costs. (ii) These results allow us to formulate a *highly efficient active-set algorithm* that can deal with input dimensions in the millions for all p-norms. This efficient algorithm enables us to directly compare the prediction performance and interpretability of solutions for all different p-norms. (iii) The solution path can be approximated on a fixed grid of constraint values with almost no additional computational costs.

Large-scale applications using both synthetic and real data illustrate the excellent performance of the developed concepts and algorithms. In particular, we demonstrate that the proposed completeness test successfully detects ambiguous solutions and thus avoids the misinterpretation of "identified" explanatory factors.

For the comparison of the different Group-Lasso methods, we consider two common application scenarios of the Group-Lasso. On the one hand, the Group-Lasso is used as a generalization of the standard Lasso for prediction problems in which single explanatory factors are encoded by a group of variables. Examples of this kind include dummy coding for categorical measurements or polynomial expansions of input features. In these cases, the focus is on *interpretation*, since it may be difficult to interpret a solution which is sparse on the level of single variables.

On the other hand, the Group-Lasso is often used in *multi-task* learning problems, as explained in Section 3.6, where the likelihood *factorizes* over the individual tasks. The motivation for using the Group-Lasso is to *couple* the individual tasks via the group-structure of the constraint term. Multi-task learning is based on the assumption that multiple tasks share some features or structures. Each task should benefit from the information content of data of all the other tasks, so that many learning problems can be solved in parallel, as was shown in [Argy 07]. It should be noticed that in this case the Group-Lasso cannot be interpreted as a direct generalization of the standard Lasso, since the latter is unable to couple the individual tasks.

The remainder of this chapter is organized as follows: In Section 4.1, conditions for the completeness and uniqueness of all $\ell_{1,p}$ Group-Lasso estimates and a simple procedure for testing for uniqueness are given. In Section 4.2, an active set algorithm is derived that is able to deal with input dimensions in the millions so that large-scale problems can be handled efficiently. In Sections 4.3 and 4.4 we report experiments on simulated and real data sets which demonstrate the behavior of the different $\ell_{1,p}$ Group-Lasso methods.

4.1 Characterization of Solutions for the $\ell_{1,p}$ Group-Lasso

In this Section we follow the main ideas as [Osbo 00], with the difference that we deal with the $\ell_{1,p}$ Group-Lasso and with a more general class of likelihood functions from the exponential family of distributions, the generalized linear models, as introduced in Section 3.2. Theoretical aspects of the $\ell_{1,2}$ Group-Lasso have been investigated analogously in [Roth 08]. Our derivations in this section follow the approach in [Roth 08] closely. Our genuine contribution consists in the characterization of solutions for *all* p-norms instead of the limited case of solely the 2-norm.

On input we are given an i.i.d. data sample $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, $\boldsymbol{x}_i \in \mathbb{R}^d$, arranged as rows of the data matrix X. The rows of X denote the observations that were made, for instance patients in a medical survey or number of measured chips in a gene expression experiment. The columns of X denote the variates such as age or weight of patients, or the genes that were measured and so on. The set of covariates or explanatory factors is arranged as the $n \times d$ matrix X. By the column vector $\boldsymbol{y} = (y_1, \ldots, y_n)^T$ we denote a corresponding vector of responses.

In the following, we will consider the problem of minimizing the negative log-likelihood

$$l(\boldsymbol{y}, \boldsymbol{\eta}, \phi) = -\sum_i \log f(y_i; \eta_i, \phi) \tag{4.1}$$

where the exponential-family distribution f is the random component of a generalized linear model (GLM),

$$f(y; \eta, \phi) = \exp\left(\phi^{-1}(y\eta - b(\eta)) + c(y, \phi)\right). \tag{4.2}$$

The GLM is completed by introducing a systematic component $\eta = \boldsymbol{x}^T\boldsymbol{\beta}$ and a strictly monotone differentiable (canonical) link function specifying the relationship between the random and systematic components: $\nu(\mu) = \eta$, where $\mu = E_\eta[y]$ is related to the natural parameter η of the distribution f by $\mu = b'(\eta) = \nu^{-1}(\eta)$. As mentioned in Section 3.2, an important property of this framework is that $\log f(y; \eta, \phi)$ is strictly concave in η. For the sake of simplicity we fix the scale parameter ϕ to 1.

With $\eta = \boldsymbol{x}^T\boldsymbol{\beta}$, the gradient of $l(\boldsymbol{y}, \boldsymbol{\eta}, \phi)$ can be viewed as a function in either $\boldsymbol{\eta}$ or $\boldsymbol{\beta}$

$$\begin{aligned} \nabla_\eta l(\boldsymbol{\eta}) &= -(\boldsymbol{y} - \nu^{-1}(\boldsymbol{\eta})), \\ \nabla_\beta l(\boldsymbol{\beta}) &= X^T \nabla_\eta l(\boldsymbol{\eta}) = -X^T(\boldsymbol{y} - \nu^{-1}(X\boldsymbol{\beta})), \end{aligned} \tag{4.3}$$

where $\nu^{-1}(\boldsymbol{\eta}) := (\nu^{-1}(\eta_1), \ldots, \nu^{-1}(\eta_n))^T$. The corresponding Hessians are

$$H_\eta = W, \quad H_\beta = X^T W X, \tag{4.4}$$

where W is diagonal with elements

$$W_{ii} = (\nu^{-1})'(\eta_i) = \mu'(\eta_i) = b''(\eta_i).$$

For the following derivation, we partition X, $\boldsymbol{\beta}$ and $\boldsymbol{h} := \nabla_\beta l$ into J subgroups:

$$X = (X_1, \ldots, X_J), \quad \boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{\beta}_1 \\ \vdots \\ \boldsymbol{\beta}_J \end{pmatrix}, \quad \boldsymbol{h} = \begin{pmatrix} \boldsymbol{h}_1 \\ \vdots \\ \boldsymbol{h}_J \end{pmatrix} = \begin{pmatrix} X_1^T \nabla_\eta l \\ \vdots \\ X_J^T \nabla_\eta l \end{pmatrix}. \tag{4.5}$$

As stated above, b is strictly convex in $\boldsymbol{\eta}$, thus $b''(\eta_i) > 0$ which in turn implies that $H_\eta \succ 0$ and $H_\beta \succeq 0$. This means that l is a strictly convex

function in $\boldsymbol{\eta}$. For general matrices X it is convex in $\boldsymbol{\beta}$, and it is strictly convex in $\boldsymbol{\beta}$ if X has full rank and $d \leq n$.

In the following derivations we follow [Roth 08], with the difference that we consider *all* p-norms. Given X and $\boldsymbol{y}$, the Group-Lasso minimizes the negative log-likelihood viewed as a function in $\boldsymbol{\beta}$ under a constraint on the sum of the ℓ_p-norms of the sub-vectors $\boldsymbol{\beta}_j$:

$$\begin{aligned} l(\boldsymbol{\beta}) &\longrightarrow \min \\ \text{s.t. } g(\boldsymbol{\beta}) &\geq 0, \end{aligned} \tag{4.6}$$

$$\text{where} \quad g(\boldsymbol{\beta}) = \kappa - \sum_{j=1}^{J} \|\boldsymbol{\beta}_j\|_p \quad \text{and} \quad 1 \leq p \leq \infty. \tag{4.7}$$

Here $g(\boldsymbol{\beta})$ is implicitly a function of the fixed parameter κ.

Considering the *unconstrained* problem, the solution is not unique if the dimensionality exceeds n: every $\boldsymbol{\beta}^* = \boldsymbol{\beta}^0 + \boldsymbol{\xi}$ with $\boldsymbol{\xi}$ being an element of the null space $\boldsymbol{N}(X)$ is also a solution. By defining the unique value

$$\kappa_0 := \min_{\boldsymbol{\xi} \in \boldsymbol{N}(X)} \sum_{j=1}^{J} \|\boldsymbol{\beta}_j^0 + \boldsymbol{\xi}_j\|_p, \tag{4.8}$$

we will require that the constraint is active i.e. $\kappa < \kappa_0$. Note that the minimum κ_0 is unique, even though there might exist several vectors $\boldsymbol{\xi} \in \boldsymbol{N}(X)$ which attain this minimum. Enforcing the constraint to be active is essential for the following characterization of solutions. Although it might be infeasible to ensure this activeness by computing κ_0 and selecting κ accordingly, practical algorithms will not suffer from this problem: given a solution, we can always check if the constraint was active. If this was not the case, then the uniqueness question reduces to checking if $d \leq n$ (if X has full rank). In this case the solutions are usually not sparse, because the feature selection mechanism has been switched off. To produce a sparse solution, one can then try smaller κ-values until the constraint is active. In Section 4.2 we propose a more elegant solution to this problem in the form of an algorithm that approximates the solution path, i.e. the evolution of the group norms when relaxing the constraint. This algorithm can be initialized with an arbitrarily small constraint value κ^0 which typically ensures that the constraint is active in the first optimization step. Activeness of the constraint in the following steps can then be monitored by observing the decay of the Lagrange parameter when increasing κ.

We will restrict our further analysis to models with finite likelihood $f < +\infty$, i.e. $l > -\infty$, which is usually satisfied for models of practical importance (see

[Wedd 73] for a detailed discussion). Technically this means that we require that the domain of l is $\mathbb{R}^d$, which implies that Slater's condition holds.

In summary, we can state the following theorem:

Theorem 4.1.1 *If $\kappa < \kappa_0$ and X has maximum rank, then the following holds: (i) A solution $\widehat{\boldsymbol{\beta}}$ exists and $\sum_{j=1}^{J} \|\widehat{\boldsymbol{\beta}}_j\|_p = \kappa$ for any such solution. (ii) If $d \leq n$, the solution is unique.*

Proof: Under the assumption $l > -\infty$ a minimum of (4.6) is guaranteed to exist, since l is continuous and the region of feasible vectors $\boldsymbol{\beta}$ is compact. Since we assume that the constraint is active, any solution $\widehat{\boldsymbol{\beta}}$ will lie on the boundary of the constraint region. It is easily seen that $\sum_{j=1}^{J} \|\boldsymbol{\beta}_j\|_p$ is convex for $1 \leq p \leq \infty$ which implies that $g(\boldsymbol{\beta})$ is concave. Thus, the region of feasible values defined by $g(\boldsymbol{\beta}) \geq 0$ is convex. If $d \leq n$, the objective function l will be strictly convex if X has full rank, which additionally implies that the minimum is unique. □

The Lagrangian for problem (4.6) reads

$$\mathcal{L}(\boldsymbol{\beta}, \lambda) = l(\boldsymbol{\beta}) - \lambda g(\boldsymbol{\beta}). \tag{4.9}$$

For a given $\lambda > 0$, $\mathcal{L}(\boldsymbol{\beta}, \lambda)$ is a convex function in $\boldsymbol{\beta}$.

Under the assumption $l > -\infty$ a minimum is guaranteed to exist, since g goes to infinity if $\|\boldsymbol{\beta}\|_p \to \infty$.

The vector $\widehat{\boldsymbol{\beta}}$ minimizes $\mathcal{L}(\boldsymbol{\beta}, \lambda)$ iff the d-dimensional null-vector $\mathbf{0}_d$ is an element of the subdifferential $\partial_\beta \mathcal{L}(\boldsymbol{\beta}, \lambda)$.

The subdifferential is

$$\partial_\beta \mathcal{L}(\boldsymbol{\beta}, \lambda) = \nabla_\beta l(\boldsymbol{\beta}) + \lambda \boldsymbol{v} = X^T \nabla_\eta l(\boldsymbol{\eta}) + \lambda \boldsymbol{v}, \tag{4.10}$$

with $\boldsymbol{v} = (\boldsymbol{v}_1, \ldots \boldsymbol{v}_J)^T$ defined by

$$\|\boldsymbol{v}_j\|_q \leq 1 \quad \text{if} \quad \|\boldsymbol{\beta}_j\|_p = 0 \tag{4.11}$$

and

$$\|\boldsymbol{v}_j\|_q = 1 \quad \text{if} \quad \|\boldsymbol{\beta}_j\|_p > 0, \tag{4.12}$$

where $\frac{1}{p} + \frac{1}{q} = 1$ for $1 < p < \infty$ and if $p = 1$, then $q = \infty$ and vice versa.

Thus, $\widehat{\boldsymbol{\beta}}$ is a minimizer for fixed λ iff

$$\mathbf{0}_d = X^T \nabla_\eta l(\boldsymbol{\eta})|_{\eta=\widehat{\eta}} + \lambda v, \quad (\text{with } \widehat{\boldsymbol{\eta}} = X\widehat{\boldsymbol{\beta}}). \tag{4.13}$$

Let d_j denote the dimension of the j-th sub-vector $\boldsymbol{\beta}_j$ (i.e. the size of the j-th subgroup). Hence, for all j with $\widehat{\boldsymbol{\beta}}_j = \mathbf{0}_{d_j}$ it holds that

$$\lambda \geq \|X_j^T \nabla_\eta l(\boldsymbol{\eta})|_{\eta=\widehat{\eta}}\|_q. \tag{4.14}$$

This yields:

$$\lambda = \max_j \|X_j^T \nabla_\eta l(\boldsymbol{\eta})|_{\eta=\widehat{\eta}}\|_q \tag{4.15}$$

For all j with $\widehat{\boldsymbol{\beta}}_j \neq \mathbf{0}_{d_j}$ it holds that

$$\lambda = \|X_j^T \nabla_\eta l(\boldsymbol{\eta})|_{\eta=\widehat{\eta}}\|_q \,. \tag{4.16}$$

Lemma 4.1.2 *Let $\widehat{\boldsymbol{\beta}}$ be a solution of (4.6). Let $\lambda = \lambda(\widehat{\boldsymbol{\beta}})$ be the associated Lagrangian multiplier. Then λ and $\widehat{\boldsymbol{h}} = \nabla_\beta l(\boldsymbol{\beta})|_{\beta=\widehat{\beta}}$ are constant across all solutions $\widehat{\boldsymbol{\beta}}_{(i)}$ of (4.6).*

Proof: Since the value of the objective function $l(\boldsymbol{\eta}_{(i)}) = l_*$ is constant across all solutions and l is strictly convex in $\boldsymbol{\eta} = X\boldsymbol{\beta}$ and convex in $\boldsymbol{\beta}$, it follows that $\widehat{\boldsymbol{\eta}}$ must be constant across all solutions $\widehat{\boldsymbol{\beta}}_{(i)}$, hence $\nabla_\beta l(\boldsymbol{\beta})|_{\beta=\widehat{\beta}} = X^T \nabla_\eta l(\boldsymbol{\eta})|_{\eta=\widehat{\eta}}$ is constant across all solutions. Uniqueness of λ follows now from (4.15). □

Theorem 4.1.3 *Let λ be the Lagrangian multiplier associated with any solution $\widehat{\boldsymbol{\beta}}$ of (4.6) and let $\widehat{\boldsymbol{h}}$ be the unique gradient vector at the optimum. Let $\mathcal{B} = \{j_1, \ldots, j_p\}$ be the unique set of indices for which $\|\widehat{\boldsymbol{h}}_j\|_q = \lambda$. Then $\widehat{\boldsymbol{\beta}}_j = \mathbf{0}_{d_j}$ $\forall j \notin \mathcal{B}$ across all solutions $\widehat{\boldsymbol{\beta}}_{(i)}$ of (4.6).*

Proof: A solution with $\widehat{\boldsymbol{\beta}}_j \neq \mathbf{0}_{d_j}$ for at least one $j \notin \mathcal{B}$ would contradict (4.16). □

Completeness of Solutions. Assume we have found a solution $\widehat{\boldsymbol{\beta}}$ of (4.6) with the set of "active" groups $\mathcal{A} := \{j : \widehat{\boldsymbol{\beta}}_j \neq \mathbf{0}\}$. If it holds that

$$\mathcal{A} = \mathcal{B} = \{j : \|\widehat{\boldsymbol{h}}_j\|_q = \lambda\},$$

then there cannot exist any other solution with an active set $\mathcal{A}'$ such that $|\mathcal{A}'| > |\mathcal{A}|$. Thus, $\mathcal{A} = \mathcal{B}$ implies that all relevant groups are contained in the solution $\widehat{\boldsymbol{\beta}}$, i. e. we cannot have overlooked other relevant groups. Hence the solution is *complete*, according to [Roth 08]. If $\mathcal{A} \neq \mathcal{B}$, then the additional elements in $\mathcal{B} \setminus \mathcal{A}$ define all possible groups that could potentially become active in alternative solutions.

Uniqueness of Solutions. Note that even if $\mathcal{A}$ is complete, it might still contain redundant groups. The question if we have found a *unique* set $\mathcal{A}$ is not answered yet. The following theorem characterizes a simple test for uniqueness under a further rank assumption of the data matrix X. With $X_{\mathcal{A}}$ we denote the $n \times s$ sub-matrix of X composed of all active groups, where $\mathcal{A}$ is the active set corresponding to some solution $\widehat{\boldsymbol{\beta}}$ of (4.6). Then the following theorem holds:

Theorem 4.1.4 *Assume that every $n \times n$ sub-matrix of X has full rank and that $\mathcal{A}$ is complete, i.e. $\mathcal{A} = \mathcal{B}$. Then, if $s \leq n$, $\widehat{\boldsymbol{\beta}}$ is the unique solution of (4.6).*

Proof: Since the set $\mathcal{B}$ is unique, the assumption $\mathcal{A} = \mathcal{B}$ implies that the search for the optimal solution can be restricted to the space $\mathbb{S} = \mathbb{R}^s$. If $s \leq n$, then the matrix $X_{\mathcal{A}}$ must have full rank by assumption. Thus, $l(\boldsymbol{\beta}_{\mathbb{S}})$ is a strictly convex function on $\mathbb{S}$ which is minimized over the convex constraint set. This implies that $\widehat{\boldsymbol{\beta}}_{\mathbb{S}}$ is the unique minimizer on $\mathbb{S}$. Since all other $\widehat{\boldsymbol{\beta}}_{j:j\notin\mathcal{A}}$ must be zero, $\widehat{\boldsymbol{\beta}}$ is unique on the whole space. □

Figure 4.1 summarizes all theoretical details of this Section in form of a flow-chart.

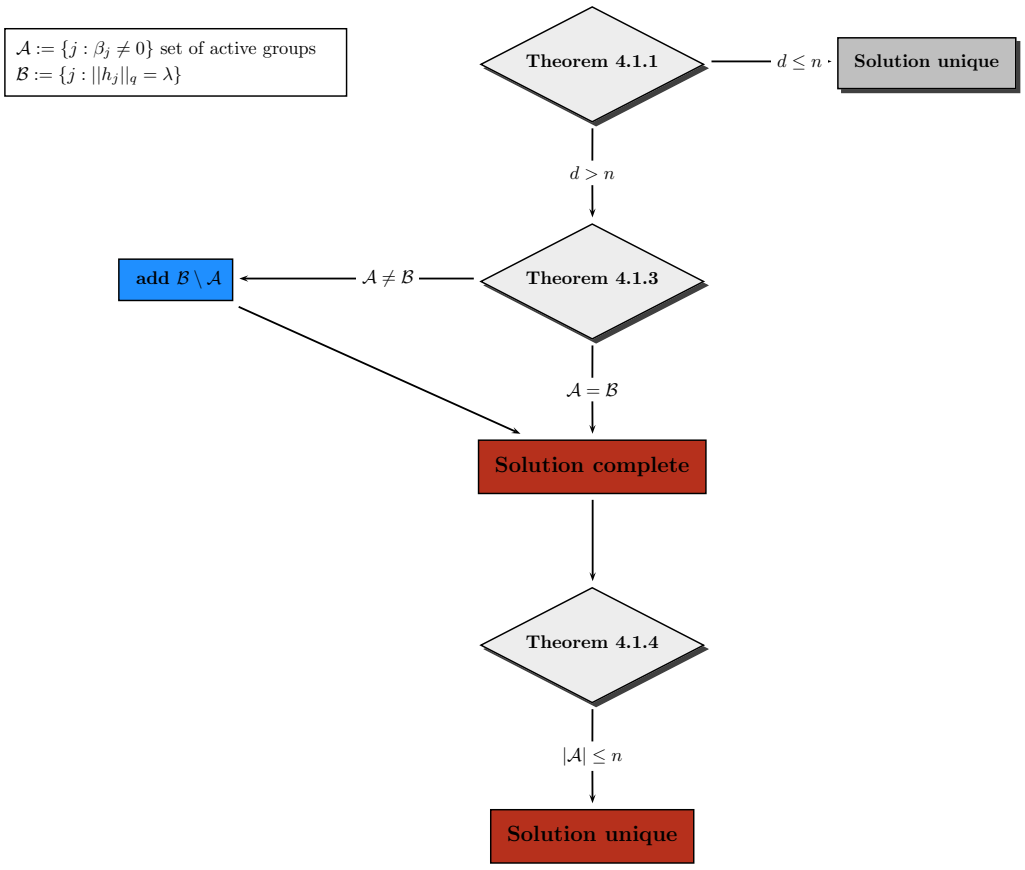

Figure 4.1: Flow-chart of the theoretical derivations of Section 4.1

4.2 An Efficient Active-Set Algorithm

The characterization of the optimal solution presented in Section 4.1 allows us to construct an active set algorithm to solve the constrained optimization problem (4.6) for all $\ell_{1,p}$-norms. The algorithm is presented in Algorithm 1. It starts with only one active group. In every iteration, further active groups are selected or removed, depending on the violation of the Lagrangian condition. The algorithm is a straightforward generalization of the subset algorithm for the standard Lasso problem presented in [Osbo 00]. The main idea is to find

a small set of active groups. Testing for completeness of the active set will then identify all groups that could have nonzero coefficients in alternative solutions.

Algorithm 1: Active Set Algorithm

A : **Initialize** set $\mathcal{A} = j_0$, $\boldsymbol{\beta}_{j_0}$ arbitrary with $\left\|\boldsymbol{\beta}_{j_0}\right\|_p = \kappa$.

B : **Optimize** over the current active set $\mathcal{A}$.

Define set $\mathcal{A}^+ = \left\{j \in \mathcal{A} : \left\|\boldsymbol{\beta}_{j_0}\right\|_p > 0\right\}$. Define $\lambda = \max_{j \in \mathcal{A}^+} \|\boldsymbol{h}_j\|_q$. Adjust the active set $\mathcal{A} = \mathcal{A}^+$.

C : **Lagrangian violation**: $\forall j \notin \mathcal{A}$, check if $\|\boldsymbol{h}_j\|_q \leq \lambda$. If this is the case, we have found a global solution. Otherwise, include the group with the largest violation to $\mathcal{A}$ and go to **B**.

D: Completeness and uniqueness. $\forall j \notin \mathcal{A}$, check if $\|\boldsymbol{h}_j\|_q = \lambda$. If so, there might exist other solutions with identical costs that include these groups in the active set. Otherwise, the active set is *complete* in the sense that it contains all relevant groups. If $X_{\mathcal{A}}$ has full rank $s \leq n$, *uniqueness* can be checked additionally via theorem 4.1.4. Note that step **D** requires (almost) no additional computations, since it is a by-product of step **C**.

Analogous to [Roth 08], Algorithm 1 can easily be extended to more practical optimization routines by stopping the fitting process at a predefined tolerance level. We can then test for completeness within a ϵ-range, i. e. $|\|\boldsymbol{h}_j\|_q - \lambda| < \epsilon$ in **D**. ϵ is defined as the maximum deviation of gradient norms from λ in the active set. This testing procedure identifies all potentially active groups in alternative solutions with costs close to the actual costs.

The optimization in step **B** can be performed by the projected gradient method ([Bert 95]). The main challenge typically is to compute efficient projections onto the $\ell_{1,p}$ ball. In general this is a hard to solve nonlinear optimization problem with nonlinear and even non-differentiable constraints. For the $\ell_{1,2}$-norm, [Kim 06] presented an efficient algorithm for the projection to the $\ell_{1,2}$ ball and the projection to the $\ell_{1,\infty}$ ball can be performed efficiently by the method introduced in [Quat 09]. The $\ell_{1,1}$ ball can be seen as a special case of the projection to the $\ell_{1,2}$ ball. An efficient projection to the $\ell_{1,p}$ ball was presented in [Liu 10a].

In general, the main idea in the projected gradient method is that one does not optimize problem (4.6) directly but solves a subproblem with quadratic cost instead. First we take a step $s\nabla_\beta l(\boldsymbol{\beta})$ along the the negative gradient with step size s and obtain the vector $\boldsymbol{b} = \boldsymbol{\beta} - s\nabla_\beta l(\boldsymbol{\beta})$. We then project $\boldsymbol{b}$ on the convex feasible region to obtain a feasible vector. Hence, the minimization problem we need to solve now reads

$$\min_{\beta} \quad \|\boldsymbol{b} - \boldsymbol{\beta}\|_2^2 + \mu \left(\sum_{j=1}^{J} \|\boldsymbol{\beta}_j\|_p - \kappa \right) \tag{4.17}$$

with Lagrangian multiplier μ. Algorithm 2 shows the projection for all $\ell_{1,p}$ norms with $1 < p < \infty$.

Algorithm 2: Optimization Step **B** for $p \in (1, \infty)$

B1 : **Gradient** :
At time $t-1$, set $\boldsymbol{b} = \boldsymbol{\beta}^{t-1} - s\nabla_\beta l(\boldsymbol{\beta}^{t-1})$ and $\mathcal{A}^+ = \mathcal{A}$, where s is the step size parameter.
Initialize Lagrangian multiplier μ within the interval $(0, \mu_{\max})$.

B2 : **Projection** :
For all $j \in \mathcal{A}^+$ minimize (4.17):
while $\sum_{j=1}^{J} \|\boldsymbol{\beta}_j^t\|_p \neq \kappa$ **do**
  $\tilde{\mathbf{B}}$: *Compute* projection as in [Liu 10a]:
  for $j \in \mathcal{A}^+$ **do**
    solve $\min_{\beta_j} \quad \|\boldsymbol{b}_j - \boldsymbol{\beta}_j\|_2^2 + \mu\|\boldsymbol{\beta}_j\|_p$:
    Compute c^*, the unique root of $\phi(c) = \mu\psi(c) - c, c \geq 0$ where $\psi(c) = ||\omega^{-1}(c)||_p^{1-p}$ and $\omega_i^{-1}(c)$ is the inverse function of $\omega_i(x) = (b_{ji} - x)/x^{p-1}, 0 < x \leq b_{ji}$ for $i = 1, .., d_j$.
    Obtain optimal $\boldsymbol{\beta}_j^*$ as the unique root of $\varphi_{c^*}^b$ where $\varphi_c^b(\boldsymbol{x}) = \boldsymbol{x} + c\boldsymbol{x}^{(p-1)} - \boldsymbol{b}, \mathbf{0} < \boldsymbol{x} < \boldsymbol{b}$.
  Adapt Lagrangian multiplier μ via interval bisection.

B3 : **New solution**: $\forall j \in \mathcal{A}^+$, set $\boldsymbol{\beta}_j^t = \boldsymbol{\beta}_j^*$

Note that the projection to the $\ell_{1,1}$ ball can be seen as a special case of the projection to the $\ell_{1,2}$ ball, hence one can use Algorithm 2 for these cases as well. The only case that has to be handled separately is the projection to the $\ell_{1,\infty}$ ball, which is given in Algorithm 3.

Algorithm 3: Optimization Step **B** for $p = \infty$

begin

B1 : **Gradient**: At time $t-1$, set $\boldsymbol{b}^* := \boldsymbol{\beta}^{t-1} - s\nabla_\beta l(\boldsymbol{\beta}^{t-1})$ where s is the step size parameter, $\mathcal{A}^+ = \mathcal{A}$ and $b_{ji} := |b^*_{ji}|$ for $i = 1, ..., d_j$.

B2 : **Projection**: Calculate vector $\boldsymbol{\theta} = (\theta_1, ..., \theta_J)$ according to [Quat 09].

B3 : **New solution** :

if $b_{ji} \geq \theta_{ji}$ then $\beta^t_{ji} = \theta_{ji}$;

if $b_{ji} \leq \theta_{ji}$ then $\beta^t_{ji} = b_{ji}$;

if $\theta_{ji} = 0$ then $\beta^t_{ji} = 0$.

B4 : **Recover sign**: $\text{sgn}(\beta^t_{ji}) := \text{sgn}(b^*_{ji})$

end

During the whole active set algorithm, access to the full set of variables is only necessary in two steps which are outside the core optimization routine, that is in steps **C** and **D**. As the need to access to all variables is outside the main optimization, Algorithm 1 is rather efficient in large-scale applications.

Note that the Group-Lasso does not exhibit a piecewise linear solution path. But we can still approximate the solution path by starting with a very small κ^0 and then iteratively relaxing the constraint. This results in a series of increasing values of κ^i with $\kappa^i > \kappa^{i-1}$. Completeness and uniqueness can be tested at every step i. As it holds that $\kappa^{(i)} > \kappa^{(i-1)}$, every previous solution $\boldsymbol{\beta}(\kappa^{(i-1)})$ is a feasible initial estimate. Then, to find $\boldsymbol{\beta}(\kappa^{(i)})$, usually only few further iterations are needed.

Convergence of Interval Bisection in Algorithm 2. It remains to show that the interval bisection within Algorithm 2 converges. This is our main technical contribution in this Section: the efficient combination of a constrained optimization problem with the Lagrangian form of an optimization problem. The projection algorithm proposed in [Liu 10a] needs the Lagrangian representation of the problem while we work with the constrained form in the active set algorithm. The combination of these two optimization problems is not trivial, as finding the appropriate Lagrangian multiplier μ could be arbitrarily sensitive to the step length s what leads to extremely slow convergence of the algorithm. Our contribution is to show that we can

combine these two methods by using an interval bisection for finding the Lagrangian multiplier μ that is guaranteed to converge rapidly.

Theorem 4.2.1 *The interval bisection in Algorithm 2 is guaranteed to converge.*

To prove Theorem 4.2.1, we first need the following Lemma.

Lemma 4.2.2 *Suppose two Lagrangian functions*

$$\mathcal{L}_1(\boldsymbol{\beta}, \mu_1) := f(\boldsymbol{\beta}) + \mu_1(\sum_{j=1}^{J} \left\|\boldsymbol{\beta}_j\right\|_p - \kappa_1) \tag{4.18}$$

$$\mathcal{L}_2(\boldsymbol{\beta}, \mu_2) := f(\boldsymbol{\beta}) + \mu_2(\sum_{j=1}^{J} \left\|\boldsymbol{\beta}_j\right\|_p - \kappa_2) \tag{4.19}$$

with convex function f, Lagrangian multipliers μ_1 and $\mu_2 \in \mathbb{R}_+$ and parameters κ_1 and $\kappa_2 \in \mathbb{R}_+$. Then, it holds that: $\mu_1 < \mu_2 \iff \kappa_2 < \kappa_1$.

Before we prove Lemma 4.2.2, we first remind of some basics of perturbation and sensitivity analysis, see e.g. [Fors 10] or [Bert 95] for more details.

In the following, let f and g denote convex functions and assume that Slater's constraint qualification is fulfilled. Consider the primal problem (P):

$$(P) \quad \begin{cases} f(\boldsymbol{\beta}) \longrightarrow \min\limits_{\beta} \\ g(\boldsymbol{\beta}) \leq 0 \end{cases}$$

The Lagrangian function $\mathcal{L}$ to (P) is defined by

$$\mathcal{L}(\beta, \mu_1) := f(\boldsymbol{\beta}) + \mu_1 g(\boldsymbol{\beta}) \tag{4.20}$$

with Lagrangian multiplier μ_1 and the dual function to (P) reads

$$\varphi(\mu_1) := \inf_{\beta} \mathcal{L}(\boldsymbol{\beta}, \mu_1). \tag{4.21}$$

The dual problem (D) to (P) has the following form:

$$(D) \quad \begin{cases} \varphi(\mu_1) \longrightarrow \max\limits_{\mu_1} \\ \mu_1 \geq 0. \end{cases}$$

Now, consider the "perturbed" primal problem (P_u) for $u \in \mathbb{R}$

$$(P_u) \quad \begin{cases} f(\boldsymbol{\beta}) \longrightarrow \min\limits_{\beta} \\ g(\boldsymbol{\beta}) \leq u. \end{cases}$$

The Lagrangian to (P_u) is the following:

$$\mathcal{L}(\boldsymbol{\beta}, \mu_2) := f(\boldsymbol{\beta}) + \mu_2(g(\boldsymbol{\beta}) - u) \tag{4.22}$$

with Lagrangian multiplier μ_2 and the dual function to (P_u) reads

$$\varphi(\mu_2) := \inf_{\beta} \mathcal{L}(\boldsymbol{\beta}, \mu_2). \tag{4.23}$$

The dual problem (D_u) to (P_u) is

$$(D_u) \quad \begin{cases} \varphi(\mu_2) \longrightarrow \max\limits_{\mu_2} \\ \mu_2 \geq 0. \end{cases}$$

Let $p(0)$ denote the optimal value for (P) and $p(u)$ the optimal value for (P_u) and μ_1 resp. μ_2 the corresponding Lagrangian multipliers, i.e.,

$$p(0) = \inf_{\beta}\{f(\boldsymbol{\beta}) + \mu_1 g(\boldsymbol{\beta})\} \tag{4.24}$$

$$p(u) = \inf_{\beta}\{f(\boldsymbol{\beta}) + \mu_2(g(\boldsymbol{\beta}) - u)\} \tag{4.25}$$

As the Slater constraint qualification is fulfilled and the problem is convex strong duality holds. This implies that μ_1 denotes the dual optimal solution to (D) and μ_2 the dual optimal solution to (D_u).

With these derivations we can now prove Lemma 4.2.2:

Proof: In the derivations above, for the choice of $g(\boldsymbol{\beta}) := \sum_{j=1}^{J} \left\|\boldsymbol{\beta}_j\right\|_p - \kappa_1$ and $u := \kappa_2 - \kappa_1$ we obtain the constraint $\sum_{j=1}^{J} \left\|\boldsymbol{\beta}_j\right\|_p \leq \kappa_1$ in problem (P) and $\sum_{j=1}^{J} \left\|\boldsymbol{\beta}_j\right\|_p \leq \kappa_2$ in problem (P_u). Hence, it holds that

$$\begin{aligned} p(0) - p(u) &= \inf_{\beta}\{f(\boldsymbol{\beta}) + \mu_1 g(\boldsymbol{\beta})\} - \inf_{\beta}\{f(\boldsymbol{\beta}) + \mu_2(g(\boldsymbol{\beta}) - u)\} \\ &= \inf_{\beta}\{f(\boldsymbol{\beta}) + \mu_1 g(\boldsymbol{\beta})\} - \inf_{\beta}\{f(\boldsymbol{\beta}) + \mu_2(g(\boldsymbol{\beta}) - \kappa_2 + \kappa_1)\} \\ &= \inf_{\beta}\{f(\boldsymbol{\beta}) + \mu_1 g(\boldsymbol{\beta})\} - \inf_{\beta}\{f(\boldsymbol{\beta}) + \mu_2 g(\boldsymbol{\beta})\} + \mu_2(\kappa_2 - \kappa_1) \\ &\geq \mu_2(\kappa_2 - \kappa_1) \end{aligned}$$

The last inequality follows because μ_1 is the optimum for (D), i.e.,

$$\inf_{\beta}\{f(\boldsymbol{\beta}) + \mu_1 g(\boldsymbol{\beta}))\} \geq \inf_{\beta}\{f(\boldsymbol{\beta}) + \mu_2 g(\boldsymbol{\beta})\}$$

On the other hand,

$$\begin{aligned} & p(0) - p(u) = \inf_{\beta}\{f(\boldsymbol{\beta}) + \mu_1 g(\boldsymbol{\beta})\} - \inf_{\beta}\{f(\boldsymbol{\beta}) + \mu_2 (g(\boldsymbol{\beta}) - u)\} \\ = & \inf_{\beta}\{f(\boldsymbol{\beta}) + \mu_1 (g(\boldsymbol{\beta}) - u)\} - \inf_{\beta}\{f(\boldsymbol{\beta}) + \mu_2 (g(\boldsymbol{\beta}) - u)\} + \mu_1 u \\ = & \inf_{\beta}\{f(\boldsymbol{\beta}) + \mu_1 (g(\boldsymbol{\beta}) - u)\} - \inf_{\beta}\{f(\boldsymbol{\beta}) + \mu_2 (g(\boldsymbol{\beta}) - u)\} + \mu_1(\kappa_2 - \kappa_1) \\ \leq & \mu_1(\kappa_2 - \kappa_1) \end{aligned}$$

The last inequality follows because μ_2 is the optimum for (D_u), i.e.,

$$\inf_{\beta}\{f(\boldsymbol{\beta}) + \mu_1 (g(\boldsymbol{\beta}) - u)\} \leq \inf_{\beta}\{f(\boldsymbol{\beta}) + \mu_2 (g(\boldsymbol{\beta}) - u)\}$$

This yields

$$\mu_2(\kappa_2 - \kappa_1) \leq p(0) - p(u) \leq \mu_1(\kappa_2 - \kappa_1)$$

Hence, we have

$$\kappa_2 < \kappa_1 \iff \mu_1 < \mu_2$$

□

With these results we now present a proof for Theorem 4.2.1:

Proof: Let

$$\tilde{g}(\mu) := \sum_{j=1}^{J} \left\|\boldsymbol{\beta}_j(\mu)\right\|_p - \kappa.$$

We denote with $\boldsymbol{\beta}(\mu) := \arg\min_{\beta} \mathcal{L}(\boldsymbol{\beta}, \mu)$ the optimal $\boldsymbol{\beta}$ for the Lagrangian function $\mathcal{L}(\boldsymbol{\beta}, \mu)$ as defined in Lemma 4.2.2. Then we get with Lemma 4.2.2 and because we know that the solution lies on the boundary of the feasible set for $\mu_1 < \mu_2$:

$$\tilde{g}(\mu_1) = \underbrace{\sum_{j=1}^{J} \left\|\boldsymbol{\beta}_j(\mu_1)\right\|_p}_{=\kappa_1} - \kappa > \underbrace{\sum_{j=1}^{J} \left\|\boldsymbol{\beta}_j(\mu_2)\right\|_p}_{=\kappa_2} - \kappa = \tilde{g}(\mu_2).$$

Hence $\tilde{g}$ is a monotonically decreasing function in the interval $[0, \mu_{\max}]$ where $\mu_{\max} := \|\boldsymbol{\beta}\|_q$ (see [Liu 10a] for details about $\mu_{\max}$). For $f(\boldsymbol{\beta}) := \|\boldsymbol{b} - \boldsymbol{\beta}\|_2^2$ it holds that

$$\tilde{g}(0) = \sum_{j=1}^{J} \|\boldsymbol{b}_j\|_p - \kappa > 0,$$

since we assume that the constraint is active. Further it holds that (see [Liu 10a], Theorem 1)

$$\tilde{g}(\mu_{\max}) = \sum_{j=1}^{J} \|\mathbf{0}\|_p - \kappa < 0.$$

In addition, $\tilde{g}$ is a continuous function, what we prove by contradiction: Assume there exists a step discontinuity that crosses zero, i.e.

$$\nexists \mu : \tilde{g}(\mu) = 0, \quad \text{and hence} \quad \nexists \mu : \sum_{j=1}^{J} \left\|\boldsymbol{\beta}_j(\mu)\right\|_p = \kappa.$$

This, however, would contradict Theorem 4.1.1, hence $\tilde{g}$ must be continuous.

According to the Intermediate Value Theorem, $\tilde{g}(\mu)$ has a unique root in $(0, \mu_{\max})$, hence the interval bisection converges. □

After each iteration of the bisection method, the size of the interval that brackets the root decreases by a factor of two. As the interval bisection is guaranteed to converge, we know that we will achieve a solution within a pre-defined tolerance interval in a logarithmic number of iterations (see e.g. [Pres 07] for more details). The convergence of the active set algorithm follows immediately:

Theorem 4.2.3 *The active set algorithm (Algorithm 1) is guaranteed to converge.*

Proof: If an obtained solution is not optimal, the solution of the augmented system will be a descent direction for the augmented problem and also for the whole problem, as primal feasibility is maintained and the constraint qualifications are fulfilled. This implies that the algorithm as a whole converges. □

4.3 Multi-Task Applications

By using the efficient unified active set algorithm which we have presented in Section 4.2, we are now able to experimentally compare the prediction performance of all p-norms for large scale experiments with thousands of features.

We address the problem of learning classifiers for a large number of tasks. In transfer or multi-task learning, we want to improve the generalization ability by solving many learning problems in parallel. Each task should benefit from the amount of information that is shared by all tasks, and such transfer learning is expected to yield better results. The motivation for using the Group-Lasso in problems of this kind is to *couple* the individual tasks via the group structure of the constraint term, based on the assumption that multiple tasks share a common sparsity pattern. Due to our efficient active set algorithm we are now able to handle data sets with thousands of features in reasonable time.

4.3.1 Synthetic Experiments.

The synthetic data for a classification problem was created in the following way: we consider a multi-task setting with m tasks and d features ($\widehat{=}$ d groups) with a $d \times m$ parameter matrix $B = [\boldsymbol{\beta}^1, ..., \boldsymbol{\beta}^m]$, where $\boldsymbol{\beta}^i \in \mathbb{R}^d$ denotes a parameter vector for the i-th task. Further, we assume we have a data set $D = (z_1, ..., z_n)$ with points z belonging to some set Z, where Z is the set of tuples $(\boldsymbol{x}_i, y_i, l_i)$ for $i = 1, ..., n$. Each $\boldsymbol{x}_i \in \mathbb{R}^d$ is a feature vector, $l_i \in 1, ..., m$ is a label that specifies to which of the m tasks the example belongs to and $y_i \in \{-1, 1\}$ is the corresponding class label. First, we generated the parameter matrix B by sampling each entry from a normal distribution $\mathcal{N}(0, 1)$. We selected 2% of the features to be the set V of relevant features and zeroed the other matrix entries.

We ran four rounds of experiments where we changed the shared sparsity pattern across the different tasks. In the first round all tasks have exactly the same sparsity pattern, just the values of $\boldsymbol{\beta}^i$ differ. In the second experiment, the tasks share 75% of the sparsity pattern, in the third experiment 50% and in the last experiment only 30%. For the training set, we sampled n-times a $d \times m$ matrix, where each entry of the matrix was sampled from the normal distribution $\mathcal{N}(0, 1)$. The corresponding labels $\boldsymbol{y} \in \mathbb{R}^{nm}$ are computed by $\boldsymbol{y}^k = (\text{sgn}((\boldsymbol{\beta}^k)^T \boldsymbol{x}_1^k), ..., \text{sgn}((\boldsymbol{\beta}^k)^T \boldsymbol{x}_n^k))^T \in \mathbb{R}^n$ for $k = 1, ..., m$. The test data

was obtained by splitting the training data in three parts, a training set, a validation set used for model selection in the cross-validation loop and an "out-of-bag" set used as a final test set. We fixed the number of tasks m to 50, the number of features d to 500 and the number of examples n per task to 200.

We compared different approaches to solve the multi-task learning problem. One approach is to pool the data, i.e. combine all tasks to one "big" task. Then, we conducted single-task learning on every task separately, and we compared different $\ell_{1,p}$ Group-Lasso methods where we used the same active set algorithm, the only difference lying in the projection step. The statistical significance of the pairwise comparisons was tested with the Kruskal-Wallis rank-sum test, and post-hoc analysis was performed using the Dunn post test with Bonferroni correction [Dunn 61].

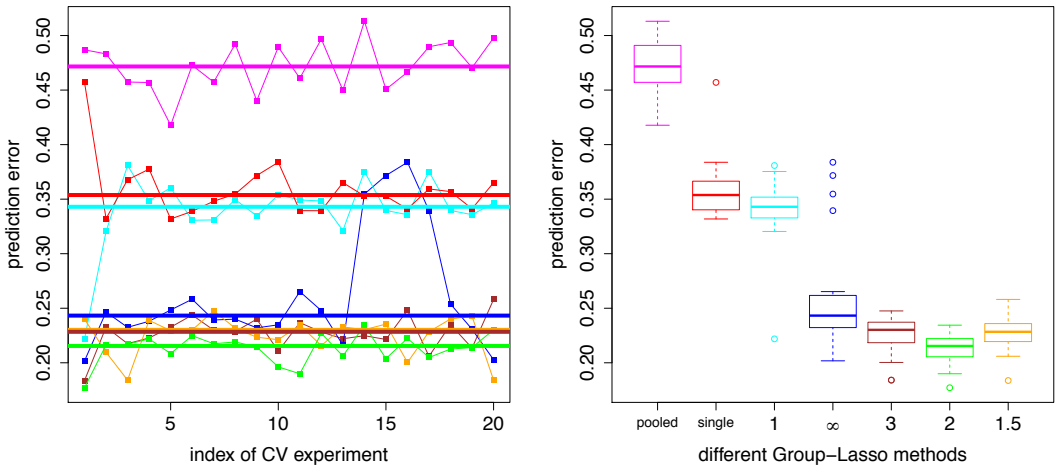

Figure 4.2: Prediction error of the different regularizers. Left panel: every cross-validation split is plotted on x-axis. Right panel: boxplot of the different Group-Lasso methods. Magenta curve and box: learning on pooled data, red curve and box: single ℓ_1, cyan curve and box: $\ell_{1,1}$, orange curve and box: $\ell_{1,1.5}$, brown curve and box: $\ell_{1,3}$, blue curve and box: $\ell_{1,\infty}$, green curve and box: $\ell_{1,2}$. In this Figure we have 100% shared sparsity pattern.

Figure 4.2 shows the result for the data set with 100% shared sparsity pattern. The left panel in Figure 4.2 displays the prediction error of the different Group-Lasso methods for every cross-validation split, whereas the right panel shows a boxplot representation of the same results. One can see that the pooled data performs worst and that single-task learning performs almost

exactly the same as the $\ell_{1,1}$ Group-Lasso. As the $\ell_{1,1}$-norm barely couples the tasks, this result is not surprising. We perceive that single-task learning is significantly worse than multi-task learning. Between all Group-Lasso methods there is no statistical significant difference. As we have exactly the same sparsity pattern in every task, even the very strong coupling of the $\ell_{1,\infty}$-norm leads to good results. In Figure 4.3 the results for 75% shared sparsity pattern are plotted.

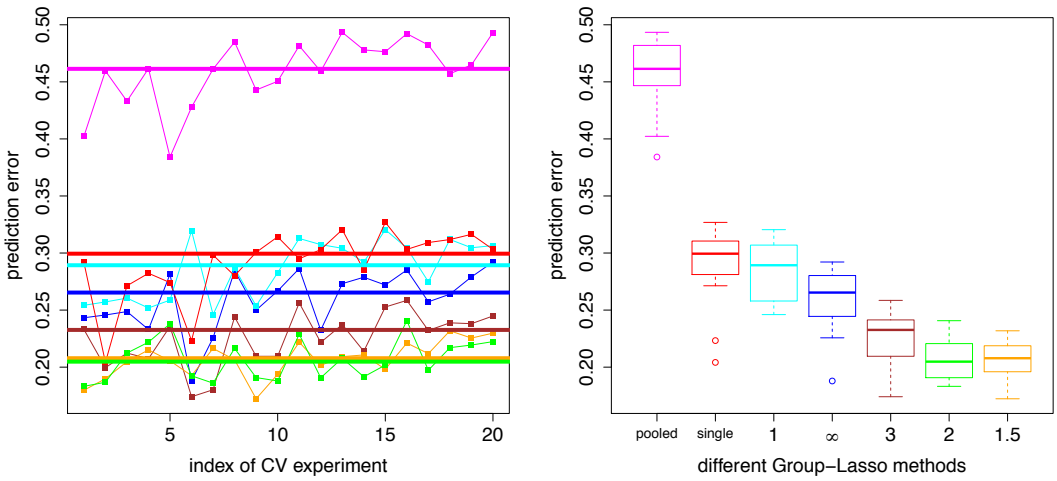

Figure 4.3: 75% shared sparsity pattern.

As in the experiment with the same sparsity pattern, pooling the data is worst and multi-task learning outperforms single-task learning. Here we can see that the strong coupling of the $\ell_{1,\infty}$-norm yields inferior results compared to the experiment before, because the sparsity pattern is not exactly the same across the different tasks anymore. There is no significant difference between the $\ell_{1,2}$-norm and the $\ell_{1,1.5}$-norm. By further reducing the joint sparsity pattern we observe that the very tight coupling of the $\ell_{1,\infty}$-norm leads to even worse results than single-task learning and we see a statistical significant advantage of the weak coupling norms $\ell_{1,2}$ and $\ell_{1,1.5}$ over all other methods, as shown in Figure 4.4. If we reduce the shared sparsity pattern to only 30%, we can nicely see that in this case the weak coupling norm $\ell_{1,1.5}$ shows a clear advantage and the strong coupling norms $\ell_{1,3}$ and $\ell_{1,\infty}$ are even worse than single-task learning. These results are collected in Figure 4.5.

In all experiments, there is not one single case where the strong coupling $\ell_{1,\infty}$-norm performs better than the weak coupling regularizations. For all values of p with $1 \leq p \leq \infty$, values for $p \in [1.5, 2]$ seem to be the best compromise between *no* coupling and very strong coupling.

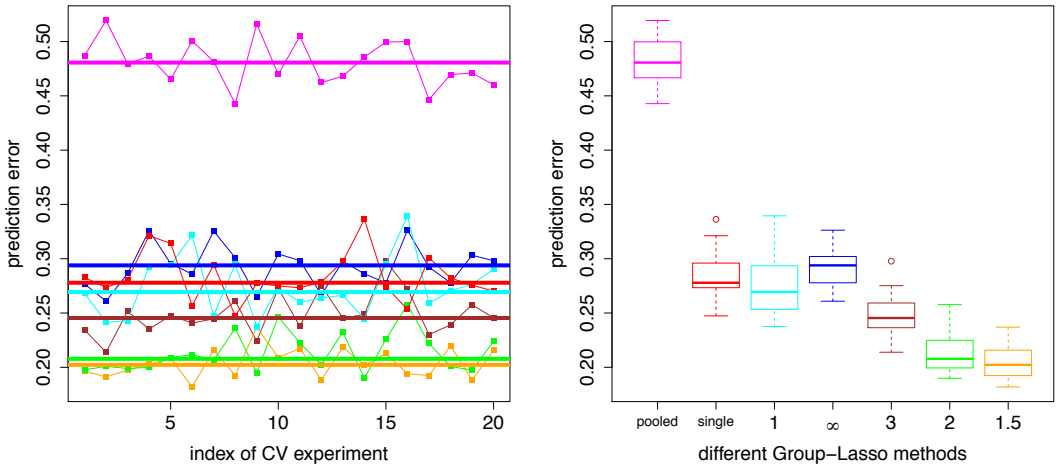

Figure 4.4: 50% shared sparsity pattern.

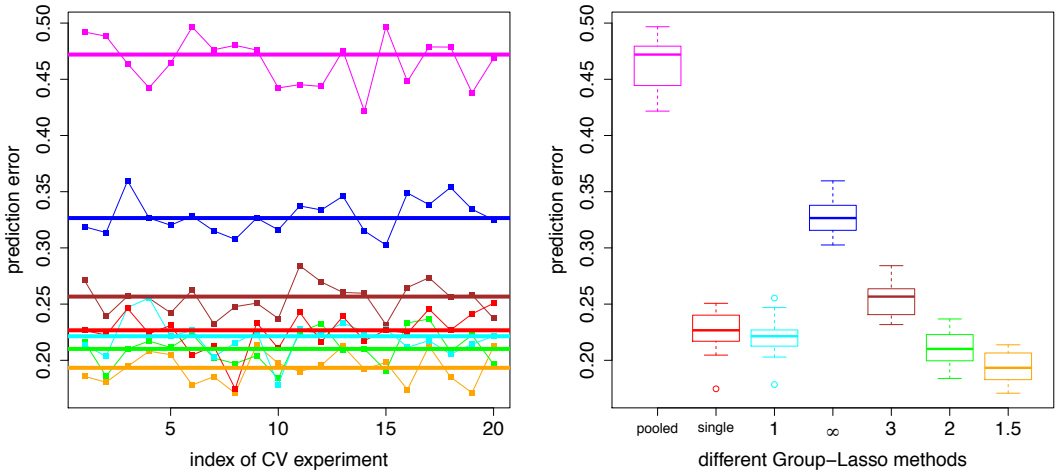

Figure 4.5: 30% shared sparsity pattern.

There exists a plausible explanation for the better overall performance of the weak coupling variants: the different tasks are connected with each other only over the constraint term. In practice, the assumption of a shared sparsity pattern among all tasks might be too restrictive, and the low-p-norms will benefit from their increased flexibility.

4.3.2 Efficiency of the Algorithm

We test the efficiency of our active set algorithm by comparing our method with the $\ell_{1,p}$-norm-regularization introduced in [Liu 10a]. To our knowledge, the method proposed in [Liu 10a] is the only existing method that can compute Group-Lasso solutions for all $\ell_{1,p}$-norms. We created synthetic data in the same way as explained in Section 4.3.1 and compared the run time of our algorithm and the algorithm proposed by [Liu 10a] for a fixed number of relevant features. The code for ([Liu 10a])'s method is publicly available[1]. The results are summarized in Figure 4.6. The dashed lines show the run time in log-log scale for the algorithm in [Liu 10a], the lines show the run time for our proposed active set algorithm. We plotted the run time for the $\ell_{1,1.5}$, $\ell_{1,3}$, $\ell_{1,\infty}$, and $\ell_{1,2}$ Group-Lasso methods in Figure 4.6. It is obvious that our active set method is significantly faster if the data set contains many groups. For [Liu 10a]'s algorithm, the steep increase between 10000 and 20000 groups is due to numerical problems in their optimizer. This comparison shows the huge advantage of using an active set method due to the explicit focus on the relatively small sect of active groups.

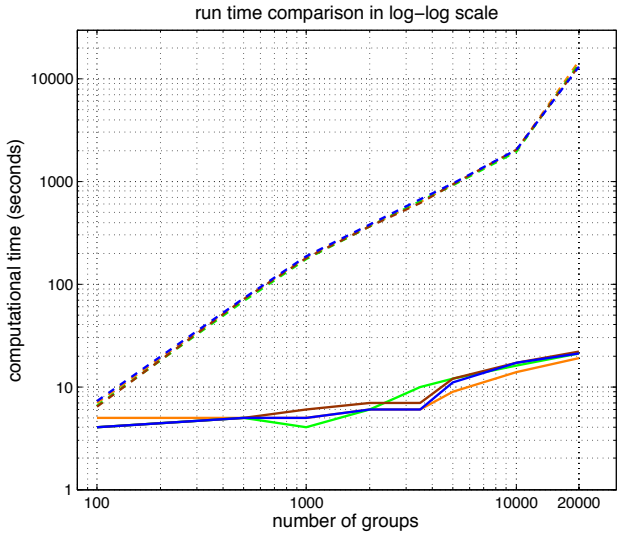

Figure 4.6: Run time in log-log-scale for our efficient active set algorithm (lines) and the algorithm proposed in [Liu 10a] (dashed lines). We plotted the run time for the $\ell_{1,1.5}$, $\ell_{1,3}$, $\ell_{1,\infty}$, and $\ell_{1,2}$ Group-Lasso methods.

[1] http://www.public.asu.edu/ jye02/Software/SLEP/index.htm

4.3.3 MovieLens Data Set

We applied different Group-Lasso methods on the MovieLens data set that was already introduced in Section 3.6. MovieLens contains 100,000 ratings for 1682 movies from 943 users.[2] The "genre" information of the movies is used as features and the ratings of the users are given on a five-point scale (1, 2, 3, 4, 5). In the terminology of multi-task learning, every user defines a task, hence we have 943 tasks in a 19-dimensional space defined by 19 different movie genres.

Similar to the synthetic experiments we compared different approaches to solve the regression problem, including single-task learning and different $\ell_{1,p}$ Group-Lasso variants. The statistical significance of differences among the pairwise comparisons was again tested with the Kruskal-Wallis rank-sum test and the Dunn post test with Bonferroni correction. From the results in Figure 4.7 we conclude that there is a statistically significant advantage of multi-task learning over single-task learning. Among the Group-Lasso methods, the very strong coupling of the $\ell_{1,\infty}$-norm yields the worst result. Between $\ell_{1,1.5}$, $\ell_{1,3}$ and the $\ell_{1,2}$ Group-Lasso there is no significant difference.

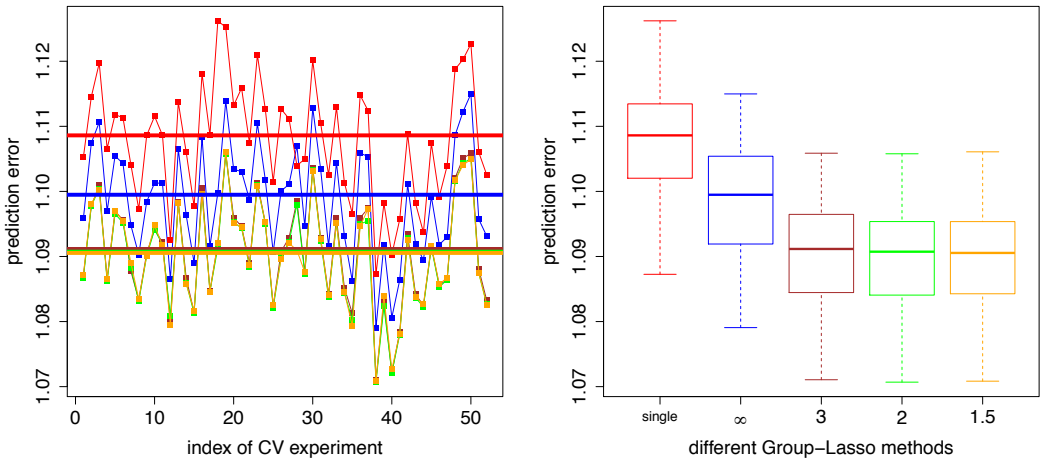

Figure 4.7: Prediction error of the different regularizers for the MovieLens data set: red curve and box: single ℓ_1, orange curve and box: $\ell_{1,1.5}$, blue curve and box: $\ell_{1,\infty}$, green curve and box: $\ell_{1,2}$.

[2]The data is available at http://www.grouplens.org.

4.3.4 Prostate Cancer Classification

A second real-world data set we looked at is a prostate cancer set that consists of two tasks. The gene measurements in either task stem from prostate tumor and non-tumor samples. The goal is to predict a patients' risk of relapse following local therapy. The idea is that by a better prediction of the outcome for men with prostate cancer, improved personalized treatment for every patient is possible.

The first data set from [Sing 02] is made up of laser intensity images from microarrays. The RMA normalization was used to produce gene expression values from these images. The second data set from [Wels 01] is already in the form of gene expression values. Although the collection techniques for both data sets were different, they share 12,600 genes which are used as features in this experiment.

We used the same experimental setup as in [Zhan 10], i.e. we used 70% of each task as training set. The results of 20 cross-validation splits are shown in Figure 4.8. Even with only two tasks, we observe that single task learning is significantly outperformed by multi-task learning. In this experiment, again the $\ell_{1,1.5}$-norm Group-Lasso yields the best result.

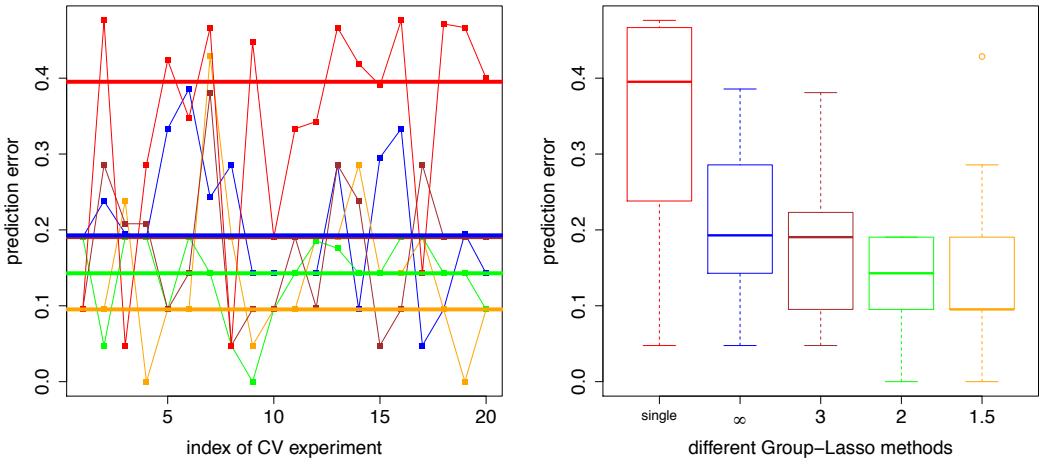

Figure 4.8: Classification error of the different Group-Lasso norms on the prostate cancer data set. Again, the $\ell_{1,1.5}$-norm (in orange) gives the best result. Single-task learning (in red) is significantly worse than all multi-task Group-Lasso methods.

4.4 Standard Prediction Problems

4.4.1 Splice Site Detection

In order to investigate the interpretability of Group-Lasso solutions, in a third real-world experiment we considered the splice site detection problem as it was discussed in [Roth 08] for the $\ell_{1,2}$ Group-Lasso. We compare the $\ell_{1,2}$ Group-Lasso with the extreme case of the $\ell_{1,\infty}$ Group-Lasso.

The prediction of splice sites plays an important role in gene finding algorithms. First, we briefly explain what splice sites are: the DNA can be seen as a long string of characters. Every character in this string is chosen from the alphabet {A,C,T,G}, like for example "ACAAGATGCCATTGTCCC" and represents a particular type of nucleic acid: A - Adenine, C-Cytosine, T-Thymine and G-Guanine. Within such long strings there are sections which are known as genes which are responsible for the creation of proteins. There exist two types of sub-sections within genes which are of special interest, the exons and the introns. Exons and introns alternate in a given DNA sequence. The role of exons is to produce proteins. Introns are the non-coding regions within a gene that separate neighboring exons. Introns always have two distinct nucleotides at either end. At the 5' end the DNA nucleotides are "GT" and at the 3' end the DNA nucleotide are "AG". A splice site is the position within a DNA that separates an intron from an exon. The 5' end of an intron is called donor splice site and the 3' end acceptor splice site. During the protein generation process, the introns are first identified and then removed. By identifying the exons and introns a problem that arises is to identify genuine splice sites from "false" splice sites.

The *MEMset Donor* dataset[3] consists of a training set of 8415 true and 179438 false human donor sites. An additional test set contains 4208 true and 89717 "false" (or *decoy*) donor sites. A sequence of a real splice site is modeled within a window that consists of the last 3 bases of the exon and the first 6 bases of the intron, c.f. Figure 4.9.

Decoy splice sites also match the consensus sequence at position zero and one. Removing this consensus "GT" results in sequences of length 7, i.e. sequences of 7 factors with 4 levels $\{A, C, G, T\}$.

The goal of this experiment is to overcome the restriction to marginal probabilities (main effects) in the widely used *Sequence-Logo* approach by exploring

[3]Available at http://genes.mit.edu/burgelab/maxent/ssdata/.

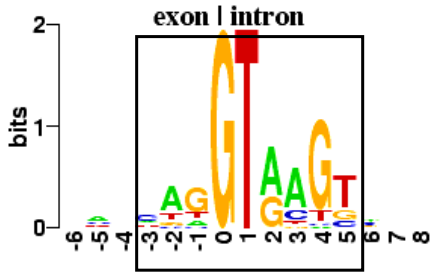

Figure 4.9: Sequence Logo representation of the human 5' splice site. The overall height of the stack of symbols at a certain position represent the sequence conservation at that position. The height of symbols within a stack represent the relative frequency of each nucleic acid. The consensus "GT" appears at position 0, 1.

all possible interactions up to order 4. Every interaction is encoded using dummy variables and treated as a group. [Roth 08] considered one experiment with a small window size and one with a bigger window size, resulting in a huge number of dimensions. We used the identical experimental setup to ensure that the results are comparable and obtained almost the same results. Contrary to the previous results in Section 4.3, in this case we see no significantly different behavior of the strong coupling $\ell_{1,\infty}$-norm and its weaker-coupling counterparts. We elaborate the results for the problem with a larger window size where the experiment shows that the interpretation of the Group-Lasso might be complicated. The problem is the discrimination between true and false splice sites at the 3' end, see Figure 4.10.

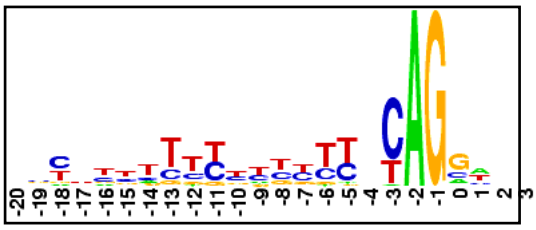

Figure 4.10: Sequence Logo representation of the human 3' splice site. The consensus "AG" appears at position -2, -1.

As in [Roth 08], we look at all interactions up to order 4, use windows of length 21 and have in total 27896 groups which span a 22,458,100-dimensional

feature space. Figure 4.11 shows our results, that are very similar to the results obtained in [Roth 08] for the $\ell_{1,2}$ Group-Lasso. For the $\ell_{1,\infty}$-norm, the optimal model at $\kappa = 60$ has correlation coefficient 0.625 (left picture of figure 4.11), compared with $\kappa = 66$ and correlation coefficient 0.631 for the $\ell_{1,2}$-norm. Hence, in terms of prediction, there is almost no difference in using the $\ell_{1,\infty}$ Group-Lasso. Among the 10 highest-scoring groups the main effects are at positions -3, -5 and 0, i.e we obtain exactly the same results as in [Roth 08]. In terms of interpretation of the solution, the $\ell_{1,\infty}$ case brings no advantage as well. The right picture in Figure 4.11 shows the results of the completeness tests. All solutions with $\kappa > 46$ are difficult to interpret, since an increasing number of groups must be added to obtain complete models. This is again almost the same result as in [Roth 08]. The number of groups that must be included in the optimal model ($\kappa = 60$) to obtain a complete model is 900, in the $\ell_{1,2}$-norm experiment the number of groups to include is 300 for the optimal $\kappa = 66$. Hence one can conclude that using the $\ell_{1,\infty}$ Group-Lasso brings no advantage, neither in terms of prediction, nor in terms of interpretability.

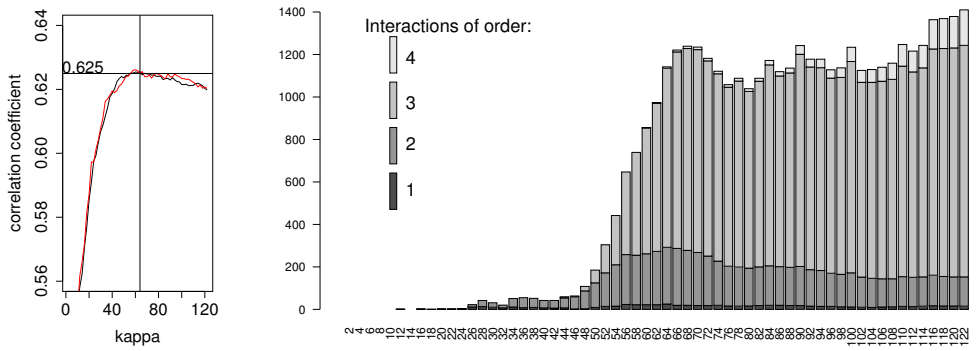

Figure 4.11: Left: Correlation coefficient as a function of κ. Red curve: correlation on the separate test set. Black curve: correlation on the validation set. Right: Acceptor splice site prediction: groups that must be included in the Group-Lasso estimates to obtain complete models (gray values stand for different orders of interactions).

4.5 Summary

We have presented a unified characterization of Group-Lasso solutions and a highly efficient active set algorithm for all $\ell_{1,p}$-variants of the Group-Lasso. With these results, we were able to directly compare all $\ell_{1,p}$ Group-Lasso methods, both in terms of prediction accuracy and interpretability of solutions in large-scale experiments. To summarize, our contribution in this chapter is threefold:

(i) On the theoretical side, we characterized conditions for solutions for all $\ell_{1,p}$ Group-Lasso methods by way of subgradient calculus. Our theoretical characterization of solutions is used to check both optimality and completeness/uniqueness.
(ii) We were able to present an active set algorithm that is applicable for all $\ell_{1,p}$ Group-Lasso methods and we proved convergence to the global optimizer. The main theoretical contribution consists in presenting a convergence proof of the interval bisection used to combine a constrained optimization problem and the Lagrangian form of an optimization problem in the inner optimization loop what leads to a fast update scheme.
(iii) On the experimental side we compared the prediction performance and the interpretability of the solutions of different Group-Lasso variants and demonstrated the efficiency of our method compared to an existing one.

We studied the interpretability of the solutions with the splice-site prediction example in a real-world context, where the inclusion of high-order factor interactions helps to increases the predictive performance but also leads to incomplete and, thus, potentially ambiguous solutions. The active set algorithm was able to approximate the solution path of the logistic Group-Lasso for feature-space dimensions up to $\approx 2 \cdot 10^7$ within a reasonable time, and the completeness test helped to avoid mis- or over-interpretations of identified interactions between the nucleotide positions. However, we could not see clear differences between the different group-norms.

The situation changes significantly when assessing the prediction performance in a multi-task setting. In a multi-task setting where the different tasks are coupled via a Group-Lasso constraint we observed clear differences in the prediction performance by using different regularizers. We examined the prediction performance of many $\ell_{1,p}$ variants and compared the different methods on synthetic data as well as on various real-world data sets.

Our experiments indicate that both the very tight coupling of the "high-p" norms with $p \gg 2$ and the too loose coupling of the "low-p" norms with $p \ll 2$ significantly degrade the prediction performance. The weak-coupling norms for $p \in [1.5, 2]$ seem to be the best compromise between coupling strength and robustness against systematic differences between the tasks.

Chapter 5

Bayesian Variable Grouping

In the previous chapters we dealt with supervised learning problems. In Chapter 4 we presented a complete analysis for the $\ell_{1,p}$ Group-Lasso and compared the prediction performance of many variants of the Group-Lasso in a multi-task learning setting, i.e., we have concentrated on supervised transfer learning so far. In the remainder of this work, we will now focus on the second aspect of this thesis, that is, on unsupervised learning problems. In this chapter, we will first present some basic background on Bayesian variable grouping before we switch to the special problem of learning on distance data directly instead on vectorial data. Then, in Chapter 6, we will present a probabilistic model that is translation- and rotation-invariant for clustering distance data. Finally, we tackle the problem of unsupervised transfer learning by extending this novel model in a way that it is able to cluster multiple views of a phenomenon.

5.1 Partition Processes

In this Section we will briefly introduce the concept of a *partition process*. Let $[n] := \{1, \ldots, n\}$ denote an index set, and $\mathbb{B}_n$ the set of partitions of $[n]$. The set $\mathbb{B}_n$ is called the "partition lattice". A partition $B \in \mathbb{B}_n$ is an equivalence relation $B : [n] \times [n] \to \{0, 1\}$ that may be represented in matrix form as

$$\begin{aligned} B(i,j) &= 1 \quad \text{if} \quad y(i) = y(j) \\ B(i,j) &= 0 \quad \text{otherwise,} \end{aligned}$$

with y being a function that maps $[n]$ to some label set $\mathbb{L}$. Alternatively, B may be represented as a set of disjoint non-empty subsets called "blocks" b. For $n \leq 4$, the sets $\mathbb{B}_n$ are the following:

$$\begin{array}{lllll} \mathbb{B}_2: & 12, & 1|2 & & \\ \mathbb{B}_3: & 123, & 12|3\,[3], & 1|2|3 & \\ \mathbb{B}_4: & 1234, & 123|4\,[4], & 12|34\,[3], & 12|3|4\,[6], \quad 1|2|2|4 \end{array}$$

With $12|3$ we denote the partition $\{\{12\},\{3\}\}$ and the number in squared brackets in $12|3\,[3]$ means that there are three partitions

$$12|3\,[3] = \{12|3, 13|2, 23|1\}.$$

The partition lattice for $\mathbb{B}_3$ is shown in Figure 5.1.

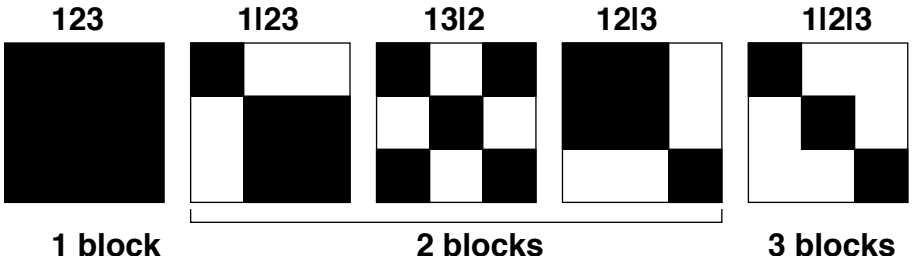

Figure 5.1: Example of the partition lattice for $\mathbb{B}_3$.

A *partition process* is a series of distributions P_n on the set $\mathbb{B}_n$ in which P_n is the marginal distribution of P_{n+1}. Such a process is called *exchangeable* if each P_n is invariant under permutations of object indices, as it was explained in detail in [Pitm 06].

5.2 Gauss-Dirichlet Clustering Process

A well-known method to partition data is the *Gauss-Dirichlet clustering process*. This process consists of an infinite sequence of points in $\mathbb{R}^d$, together with a random partition of integers into k blocks. A sequence of length n can be sampled as follows (see e.g. [MacE 94, Dahl 05, McCu 08b] for more details): fix the number of mixture modes k and generate mixing proportions $\pi = (\pi_1, \ldots, \pi_k)$ from an exchangeable Dirichlet distribution $\mathrm{Dir}(\xi/k, \ldots, \xi/k)$, generate a label sequence $\{y(1), \ldots, y(n)\}$ from a multinomial distribution and forget the labels introducing the random partition B

of $[n]$ induced by y. Integrating out π, one arrives at a Dirichlet-Multinomial prior over partitions

$$P_n(B|\xi,k) = \frac{k!}{(k-k_B)!}\frac{\Gamma(\xi)\prod_{b\in B}\Gamma(n_b+\xi/k)}{\Gamma(n+\xi)[\Gamma(\xi/k)]^{k_B}}, \tag{5.1}$$

where $k_B \leq k$ denotes the number of blocks present in the partition B and n_b is the size of block b. The limit as $k \to \infty$ is well defined and known as the Ewens process (a.k.a. Chinese Restaurant process, which was explained in Section 2.4.1). Given such a partition B, a sequence of n-dimensional observations $\boldsymbol{x}_i \in \mathbb{R}^n,\ i = 1,\ldots,d$ is arranged as columns of the $(n \times d)$ matrix X, and this X is generated from a zero-mean Gaussian distribution with covariance matrix

$$\begin{aligned} \widetilde{\Sigma}_B &= I_n \otimes \Sigma_0 + B \otimes \Sigma_1, \\ \text{with} \quad \operatorname{cov}(X_{ir}, X_{js}|B) &= \delta_{ij}\Sigma_{0rs} + B_{ij}\Sigma_{1rs}, \end{aligned} \tag{5.2}$$

where Σ_0 is the usual $(d \times d)$ "pooled" within-class covariance matrix and Σ_1 the $(d \times d)$ between-class matrix, respectively, and δ_{ij} denotes the Kronecker symbol.

Since the partition process is invariant under permutations, one can always think of B being block-diagonal. For spherical covariance matrices (i.e. scaled identity matrices), $\Sigma_0 = \alpha I_d, \Sigma_1 = \beta I_d$, the covariance structure reduces to

$$\begin{aligned} \widetilde{\Sigma}_B &= I_n \otimes \alpha I_d + B \otimes \beta I_d \\ &= (\alpha I_n + \beta B) \otimes I_d =: \Sigma_B \otimes I_d, \\ \text{with} \quad \operatorname{cov}(X_{ir}, X_{js}|B) &= (\alpha\delta_{ij} + \beta B_{ij})\delta_{rs}. \end{aligned} \tag{5.3}$$

Thus, the columns of X contain independent n-dimensional vectors $\boldsymbol{x}_i \in \mathbb{R}^n$ distributed according to a normal distribution with covariance matrix

$$\Sigma_B = \alpha I_n + \beta B.$$

Figure 5.2 shows an example of a data matrix X given a partition B, constructed in the way as described above.

Further, the distribution factorizes over the blocks $b \in B$. Introducing the symbol $i_b := \{i : i \in b\}$ defining an index-vector of all objects assigned to block b, the joint distribution reads

$$p(X, B|\alpha,\beta,\xi,k) = P_n(B|\xi,k) \cdot \left[\prod_{b\in B}\prod_{j=1}^{d}\mathcal{N}(X_{i_b j}|\alpha I_{n_b} + \beta \mathbf{1}_{n_b}\mathbf{1}_{n_b}^t)\right], \tag{5.4}$$

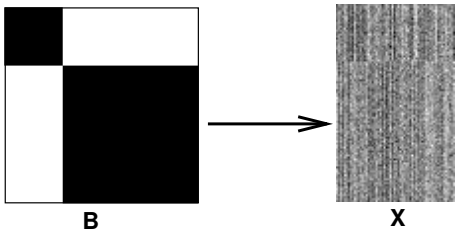

Figure 5.2: Example for a matrix X given a partition B where $X|B \sim \mathcal{N}(0, \Sigma_B)$.

where $\mathbf{1}_{n_b}$ is a n_b-vector of ones. In the following, we will use the abbreviations $\mathbf{1}_b := \mathbf{1}_{n_b}$ and $I_b := I_{n_b}$ to avoid double subscripts. Note that this distribution is expressed in terms of the partition without resorting to labels, that means that label switching cannot occur.

5.3 From Vectorial to Distance Data

Traditional machine learning methods usually depend on geometric information of the data. In the medical application example for unsupervised learning presented in Section 2.4.1, the input data was gene expression values, i.e., vectorial data. But for several applications the data is only available as scores of pairwise comparisons, since frequently no access is given to the underlying vectorial representation of the data but only pairwise similarities or distances are measured. Examples of data sets of this kind include all types of kernel matrices, be it string alignment kernels over DNA or protein sequences or diffusion kernels on graphs.

Especially in biomedical data analysis, often only distance data is available, as for instance by measuring the similarity of DNA sequences or protein sequences. One concrete example where distance data is obtained consists in the analysis of a certain type of human proteins, the so-called proteases. Proteases are cellular enzymes that conduct proteolysis, i.e. the directed degradation (digestion) of proteins. Proteases are important in a medical point of view since they play a key role in the development of metastatic tumors. To analyze proteases, the similarity of the enzymes' amino acid sequences is measured. The sequence alignment of the amino acid sequences results in a distance matrix without an underlying vectorial representation.

Pairwise data, or distance data, is in no natural way related to the common viewpoint of objects lying in some well behaved space like a vector space.

Partitioning proximity data is considered a much harder problem than partitioning vectorial data, as the inherent structure of n samples is hidden in n^2 pairwise relations. A loss-free embedding into a vector space is usually not possible. Hence, grouping problems of this kind cannot be directly transformed into a vectorial representation by means of classical embedding strategies like e.g. multi-dimensional scaling.

In the remainder of this thesis we will develop new machine learning methods based on *distance* data that do not require direct access to an underlying vector space. We propose that even if an underlying vectorial representation exists, it is better to work directly with the dissimilarity matrix to avoid unnecessary bias and variance caused by embeddings.

In Chapter 6 we introduce the translation-invariant Wishart-Dirichlet clustering process, a Bayesian clustering approach that works on distance data directly. Based on this probabilistic clustering process we then extend the model to situations when two or more views of distance data is available. This relates to the scenario of the first part of the thesis, where we considered multiple (vectorial) data sets and the aim was not to learn on every data set separately but to transfer available knowledge over related data sets and profit from the amount of data given by all data sets together. The same idea of transferring knowledge over data sets with co-occurring samples is now applied to distance data by extending the single-view learning model to a *multi-view* learning model.

Chapter 6

Translation-invariant Wishart Dirichlet Clustering Processes

The Bayesian clustering approach presented in this chapter aims at identifying subsets of objects represented as columns/rows in a dissimilarity matrix. The underlying idea is that objects grouped together in such a cluster can be reasonably well described as a homogeneous sub-population. Our focus on dissimilarity matrices implies that we do not have access to a vectorial representation of the objects. Such underlying vectorial representation may or may not exist, depending on whether the dissimilarity matrix can be embedded (without distortion) in a vector space. One way of dealing with such clustering problems would be to explicitly construct an Euclidean embedding (or possibly a distorted embedding), and to apply some more traditional clustering methods in the resulting Euclidean space. However, even under the assumption that there exists an Euclidean embedding it is better *not* to explicitly embed the data. Technically speaking, such embeddings break the symmetry induced by the translation- and rotation-invariance which reflects the information loss incurred when moving from vectors to pairwise dissimilarities. We introduce a clustering model which works directly on dissimilarity matrices. It is invariant against label- and object permutations and against scale transformations. The model is fully probabilistic in nature, which means that on output we are not given a single clustering solution, but samples from a probability distribution over partitions. If desired, a "representative" solution can be computed. Further, by using a Dirichlet process prior, the number of clusters does not need to be fixed in advance. On the algorithmic side, a highly efficient sampling algorithm is presented. Costly matrix operations are avoided by carefully exploiting the structure of the clustering problem. Invariance against label permutations is a com-

mon cause of the so-called "label switching" problem in mixture models. By formulating the model as a partition process this switching problem is circumvented.

In Section 6.2 we present a probabilistic model for combined clustering of objects that are represented via pairwise dissimilarities and occur in multiple views. In this Bayesian clustering approach, we assume the data to arrive in T different views. Each view is thought to be a conditional independent sample for one common cluster structure. The aim is to obtain a combined clustering of all views and benefit from the amount of data given by all views together. Due to its nature, the approach is permutation-, scale- and translation- invariant. As in the TIWD process, the number of clusters is inferred automatically. The advantage of this multi-view approach compared to clustering on every view separately is that one can benefit from the amount of information given by all views, in the same manner as we used this information in the supervised multi-task learning setting in Chapter 4. It might be that the cluster structure of the data is not obvious in every single view. Hence clustering on these views separately leads to poor results. But, by combining all available viewpoints one can profit from the shared structural information in the different views and hence significantly improve the cluster performance.

6.1 Wishart-Dirichlet Clustering Process

In this Section, the Gauss-Dirichlet clustering process that was introduced in Section 5.2 is extended to a sequence of inner-product and distance matrices. The underlying assumption is that the random matrix $X_{n\times d}$ follows the zero-mean Gaussian distribution specified in (5.2), with $\Sigma_0 = \alpha I_d$ and $\Sigma_1 = \beta I_d$. Then, conditioned on the partition B, the inner product matrix $S = XX^T/d$ follows a (possibly singular) Wishart distribution in d degrees of freedom, $S \sim \mathcal{W}_d(\Sigma_B)$ ([Sriv 03]). Figure 6.1 shows an example of a data matrix S given a partition B with spherical covariance matrices, i.e. the observed matrix S is explained as $\mathcal{W}_d(\Sigma_B)$ where $\Sigma_B = \alpha I + \beta B$.

If one directly observes the dot products S, it suffices to consider the conditional probability of partitions, $P_n(B|S)$, which has the same functional form for ordinary and singular Wishart distributions:

$$\begin{aligned} P_n(B|S,\alpha,\beta,\xi,k) &\propto \mathcal{W}_d(S|\Sigma_B)\cdot P_n(B|\xi,k) \\ &\propto |\Sigma_B|^{-\frac{d}{2}} \exp\left(-\frac{d}{2}\mathrm{tr}(\Sigma_B^{-1}S)\right)\cdot P_n(B|\xi,k). \end{aligned} \tag{6.1}$$

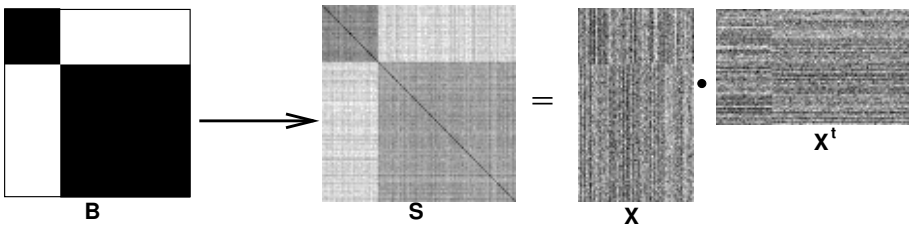

Figure 6.1: Example for a matrix S given a partition B where $S|B \sim \mathcal{W}_d(\Sigma_B)$.

In Figure 6.2 the inference of the partition B from the inner product matrix S is illustrated.

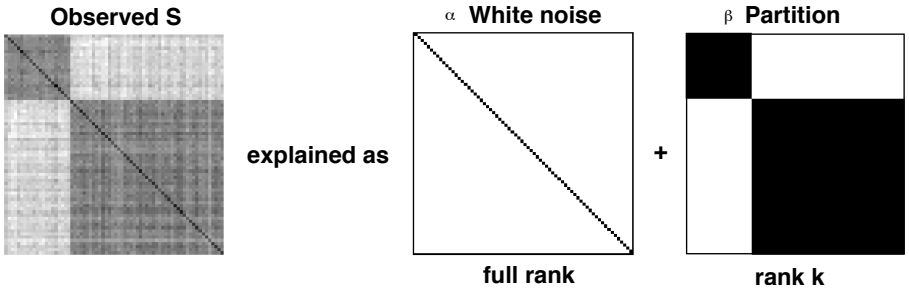

Figure 6.2: Inferring the partition B from the inner products S.

For the following derivation it is suitable to re-parametrize the model in terms of (α, θ) instead of (α, β), where $\theta := \beta/\alpha$, and in terms of $W := \Sigma_B^{-1}$. Due to the block structure in B, $P_n(B|S)$ factorizes over the blocks $b \in B$:

$$\begin{aligned} P_n(B|S, \alpha, \theta, \xi, k) &\propto P_n(B|\xi, k) \\ &\cdot \left[\prod_{b\in B} |W_b|^{\frac{d}{2}}\right] \exp\left(-\sum_{b\in B} \frac{d}{2}\mathrm{tr}(W_b S_{bb})\right), \end{aligned} \tag{6.2}$$

where W_b, S_{bb} denote the submatrices corresponding to the b-th diagonal block in B or W, as explained in Figure 6.3.

The above factorization property can be exploited to derive an efficient inference algorithm for this model. The key observation is that the inverse matrix $W_b = \Sigma_b^{-1}$ can be analytically computed as

$$W_b = (\alpha I_b + \beta \mathbf{1}_b \mathbf{1}_b^T)^{-1} = \left[\alpha(I_b + \theta \mathbf{1}_b \mathbf{1}_b^T)\right]^{-1} = \frac{1}{\alpha}\left[I_b - \frac{\theta}{1 + n_b\theta}\mathbf{1}_b\mathbf{1}_b^T\right]. \tag{6.3}$$

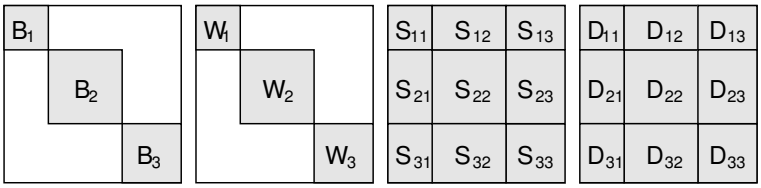

Figure 6.3: Example of the block structure of B and W (left) and the definition of sub-matrices in S and D (right) for $k_B = 3$.

Thus, the contribution of block b to the trace is

$$\mathrm{tr}(W_b S_{bb}) = \frac{1}{\alpha}\left[\mathrm{tr}(S_{bb}) - \frac{\theta}{1+n_b\theta}\bar{S}_{bb}\right], \tag{6.4}$$

where $\bar{S}_{bb} = \mathbf{1}_b^T S_{bb}\mathbf{1}_b$ denotes the sum of the b-th diagonal block of S. A similar trick can be used for the determinant which is the product of the eigenvalues: the k_B smallest eigenvalues of W are given by

$$\lambda_b = \alpha^{-1}(1+\theta n_b)^{-1}.$$

The remaining $n - k_B$ eigenvalues are equal to α^{-1}. Thus, the determinant reads

$$|W| = \prod_{b\in B}\lambda_b = \alpha^{-n}\prod_{b\in B}(1+\theta n_b)^{-1}. \tag{6.5}$$

6.1.1 Scale Invariance

The re-parametrization using (α, θ) leads to a new semantics of $(1/\alpha)$ as a scale parameter: α is excluded from the partition-dependent terms in the product over the blocks in (6.5), which implies that the conditional for the partition becomes

$$\begin{aligned} P_n(B|\bullet) \propto P_n(B|\xi,k)\cdot\left[\prod_{b\in B}(1+\theta n_b)^{-1}\right]^{d/2} \\ \cdot\exp\left(-\frac{1}{\alpha}\frac{d}{2}\sum_{b\in B}\mathrm{tr}(W_b S_{bb})\right). \end{aligned} \tag{6.6}$$

$(1/\alpha)$ simply rescales the observed matrix S, and we can make the model scale invariant by introducing a prior distribution and integrating out α. The conditional posterior for α follows an inverse Gamma distribution

$$p(\alpha|r,s) = \frac{s^r}{\Gamma(r)}\left(\frac{1}{\alpha}\right)^{r+1}\exp\left(-\frac{s}{\alpha}\right), \tag{6.7}$$

with shape parameter $r = n \cdot d/2 - 1$ and scale $s = \frac{d}{2}(\mathrm{tr}(S) - \sum_{b \in B} \frac{\theta}{1+n_b\theta} \bar{S}_{bb})$. Using an inverse Gamma prior with parameters r_0, s_0, the posterior is of the same functional form with $r_p = r + r_0 + 1$ and $s_p = s + s_0$, and we can integrate out α analytically. Dropping all terms independent of the partition structure we arrive at

$$P_n(B|\bullet) \propto P_n(B|\xi, k)|W|_{(\alpha=1)}^{d/2}(s + s_0)^{-(r+r_0+1)}, \tag{6.8}$$

where $|W|_{(\alpha=1)} = \left(\prod_{b \in B}(1 + \theta n_b)^{-1}\right)$ follows from (6.5).

6.1.2 The Centering Problem

In practice, however, there are two problems with the model described above: (i) S is often not observed directly, but only a matrix of *distances* D. In the following the assumption holds that the (suitably pre-processed) matrix D contains *squared Euclidean distances* with components $D_{ij} = S_{ii} + S_{jj} - 2S_{ij}$; (ii) even if one observes a dot-product matrix and the assumption of an underlying generative Gaussian process appears reasonable, usually no information about the mean vector $\boldsymbol{\mu}$ is given. The underlying assumption was that there exists a matrix X with $XX^T = S$ such that the *columns* of X are independent copies drawn from a zero-mean Gaussian in $\mathbb{R}^n$: $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu} = \mathbf{0}_n, \Sigma = \Sigma_B)$. This assumption is crucial, since general mean vectors correspond to a *non-central* Wishart model [Ande 46], which can be calculated analytically only in special cases, and even these cases have a very complicated form which imposes severe problems in deriving efficient inference algorithms.

Both of the above problems are related in the way that they have to do with the lack of information about geometric transformations: assume one only observes S without access to the vectorial representations $X_{n \times d}$. Then the information about orthogonal transformations $X \leftarrow XO$ with $OO^T = I_d$ is lost, i.e. there is no information about rotations and reflections of the rows in X. If only the distance matrix D is observed, one has additionally lost the information about translations of the rows, $X \leftarrow X + (\mathbf{1}_n \boldsymbol{v}^T + \boldsymbol{v}\mathbf{1}_n^T)$, with $\boldsymbol{v} \in \mathbb{R}^d$. A graphical illustration of the information loss due to rotations and translations is given in Figure 6.4.

The sampling model implies that the means in each row are expected to converge to zero as the number of replications d goes to infinity. Thus, if one had access to X and if it is not clear that the above zero-mean assumption holds, it might be a plausible strategy to subtract the empirical row means, $X_{n \times d} \leftarrow X_{n \times d} - (1/d)X_{n \times d}\mathbf{1}_d\mathbf{1}_d^T$, and then to construct a candidate

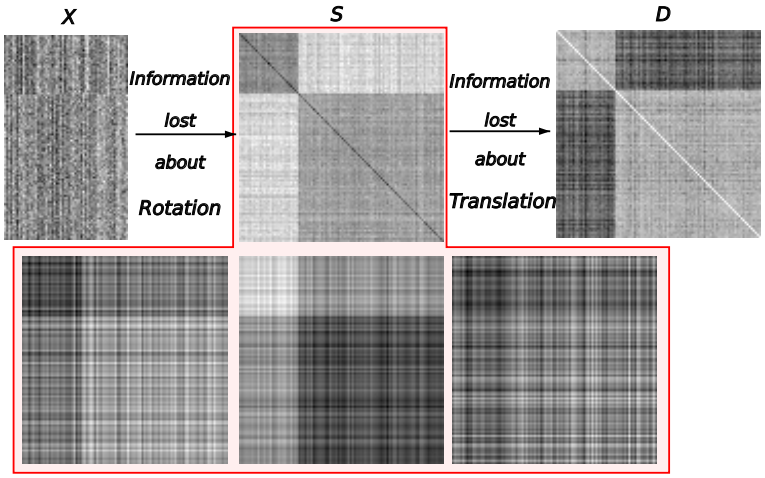

Figure 6.4: By obtaining a similarity matrix S the information about rotations and reflections of the rows of X are lost. If distances D are obtained, additionally the information about translations of the rows are lost: one D matrix leads to a whole equivalence class of S matrices, the transformation from D to S is not unique.

matrix S by computing the pairwise dot products. This procedure should be statistically robust if $d \gg n$, since then the empirical means are probably close to their expected values. Such a corrected matrix S fulfills two important requirements for selecting candidate dot product matrices: first, S should be "typical" with respect to the assumed Wishart model with $\boldsymbol{\mu} = \mathbf{0}$, thereby avoiding any bias introduced by a particular choice. Second, the choice should be robust in a statistical sense: if we are given a second observation from the same underlying data source, the two selected prototypical matrices S_1 and S_2 should be similar. For small d, this correction procedure is dangerous since it can introduce a strong bias even if the model is correct: suppose we are given two replications from $\mathcal{N}(\boldsymbol{\mu} = \mathbf{0}_n, \Sigma = \Sigma_B)$, i.e. $d = 2$. After subtracting the row means, all row vectors lie on the diagonal line in $\mathbb{R}^2$, and the cluster structure is heavily distorted.

Consider now case (ii) where we observe S without access to X. Case (i) needs no special treatment, since it can be reduced to case (ii) by first constructing a positive semi-definite matrix S which fulfills $D_{ij} = S_{ii} + S_{jj} - 2S_{ij}$. For "correcting" the matrix S just as described above we would need a procedure which effectively subtracts the empirical row means from the rows of X. Unfortunately, there exists no such matrix transformation that operates directly on S without explicit construction of X. It is important to note that the "usual" centering transformation $S \leftarrow QSQ$ with $Q_{ij} = \delta_{ij} - \frac{1}{n}$ as

used in kernel PCA and related algorithms does not work here: in kernel PCA the rows of X are assumed to be i.i.d. replications in $\mathbb{R}^d$. Consequently, the centered matrix S_c is built by subtracting the *column* means: $X_{n\times d} \leftarrow X_{n\times d} - (1/n)\mathbf{1}_n\mathbf{1}_n^T X_{n\times d}$ and $S_c = XX^T = QSQ$. Here, we need to subtract the *row* means, and therefore it is inevitable to explicitly construct X, which implies that we have to choose a certain orthogonal transformation O. It might be reasonable to consider only rotations and to use the principle components as coordinate axes. This is essentially the kernel PCA embedding procedure: compute $S_c = QSQ$ and its eigenvalue decomposition $S_c = V\Lambda V^T$, and then project on the principle axes: $X = V\Lambda^{1/2}$. The problem with this vector-space embedding is that it is statistically robust in the above sense only if d is small, because otherwise the directions of the principle axes might be difficult to estimate, and the estimates for two replicated observations might highly fluctuate, leading to different row-mean normalizations. Note that this condition for fixing the rotation contradicts the above condition $d \gg n$ that justifies the subtraction of the means. Further, row-mean normalization will change the pairwise dissimilarities, $D_{ij} = S_{ii} + S_{jj} - 2S_{ij}$, and this change can be drastic if d is small.

The cleanest solution might be to consider the dissimilarities D (which are observed in case (i) and computed as $D_{ij} = S_{ii} + S_{jj} - 2S_{ij}$ in case (ii)) as the "reference" quantity, and to avoid an explicit choice of S and X altogether. Therefore, we propose to encode the translation invariance directly into the likelihood, which means that the latter becomes constant on all matrices S that fulfill $D_{ij} = S_{ii} + S_{jj} - 2S_{ij}$.

6.1.3 The Translation-invariant WD-Process

A squared Euclidean distance matrix D is characterized by the property of being of *negative type*, which means that $\boldsymbol{x}^T D\boldsymbol{x} = -\frac{1}{2}\boldsymbol{x}^T S\boldsymbol{x} < 0$ for any $\boldsymbol{x}$ with $\boldsymbol{x}^T\mathbf{1} = 0$. This condition is equivalent to the absence of negative eigenvalues in $S_c = QSQ = -(1/2)QDQ$. The distribution of D has been formally studied in [McCu 09], where it was shown that if S follows a standard Wishart generated from an underlying zero-mean Gaussian process, $S \sim \mathcal{W}_d(\Sigma_B)$, $-D$ follows a generalized Wishart distribution, $-D \sim \mathcal{W}(\mathbf{1}, 2\Sigma_B) = \mathcal{W}(\mathbf{1}, -\Delta)$ defined with respect to the transformation kernel $\mathbb{K} = \mathbf{1}$, where $\Delta_{ij} = \Sigma_{Bii} + \Sigma_{Bjj} - 2\Sigma_{Bij}$. To understand the role of the transformation kernel it is useful to introduce the notion of a generalized Gaussian distribution with kernel $\mathbb{K} = \mathbf{1}$: $X \sim N(\mathbf{1}, \boldsymbol{\mu}, \Sigma)$. For any transformation L with $L\mathbf{1} = 0$, the meaning of the general Gaussian notation

is:

$$LX \sim \mathcal{N}(L\boldsymbol{\mu}, L\Sigma L^T). \tag{6.9}$$

It follows that under the kernel $\mathbb{K} = \mathbf{1}$, two parameter settings $(\boldsymbol{\mu}_1, \Sigma_1)$ and $(\boldsymbol{\mu}_2, \Sigma_2)$ are equivalent if $L(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) = \mathbf{0}$ and $L(\Sigma_1 - \Sigma_2)L^T = 0$, i.e. if $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 \in \mathbf{1}$, and $(\Sigma_1 - \Sigma_2) \in \{\mathbf{1}_n \boldsymbol{v}^T + \boldsymbol{v}\mathbf{1}_n^T : \boldsymbol{v} \in \mathbb{R}^n\}$, a space which is usually denoted by $\mathrm{sym}^2(\mathbf{1} \otimes \mathbb{R}^n)$. It is also useful to introduce the distributional symbol $S \sim \mathcal{W}(\mathbb{K}, \Sigma)$ for the generalized Wishart distribution of the random matrix $S = XX^T$ when $X \sim N(\mathbb{K}, \mathbf{0}, \Sigma)$. The key observation in [McCu 09] is that $D_{ij} = S_{ii} + S_{jj} - 2S_{ij}$ defines a linear transformation on symmetric matrices with kernel $\mathrm{sym}^2(\mathbf{1} \otimes \mathbb{R}^n)$ which implies that the distances follow a generalized Wishart distribution with kernel $\mathbf{1}$: $-D \sim \mathcal{W}(\mathbf{1}, 2\Sigma_B) = \mathcal{W}(\mathbf{1}, -\Delta)$. In the multi-dimensional case with spherical within- and between covariances we generalize the above model to Gaussian random matrices $X \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma_B \otimes I_d)$. Note that the d columns of this matrix are i.i.d. copies. The distribution of the matrix of squared Euclidean distances D then follows a generalized Wishart with d degrees of freedom $-D \sim \mathcal{W}_d(\mathbf{1}, -\Delta)$. This distribution differs from a standard Wishart in that the inverse matrix $W = \Sigma_B^{-1}$ is substituted by the matrix $\widetilde{W} = W - (\mathbf{1}^T W \mathbf{1})^{-1} W \mathbf{1}\mathbf{1}^T W$ and the determinant $|\cdot|$ is substituted by a generalized $\det(\cdot)$-symbol which denotes the product of the nonzero eigenvalues of its matrix-valued argument (note that $\widetilde{W}$ is rank-deficient). The conditional probability of a partition then reads

$$\begin{aligned} P(B|D, \bullet) &\propto \mathcal{W}(-D|\mathbf{1}, -\Delta) \cdot P_n(B|\xi, k) \\ &\propto \det(\widetilde{W})^{\frac{d}{2}} \exp\left(\frac{d}{4}\mathrm{tr}(\widetilde{W}D)\right) \cdot P_n(B|\xi, k). \end{aligned} \tag{6.10}$$

Note that in spite of the fact that this probability is written as a function of $W = \Sigma_B^{-1}$, it is constant over all choices of Σ_B which lead to the same Δ, i.e. independent under translations of the row vectors in X. For the purpose of inferring the partition B, this invariance property means that we can simply use our block-partition covariance model Σ_B and assume that the (unobserved) matrix S follows a standard Wishart distribution parametrized by Σ_B. We do not need to care about the exact form of S, since the conditional posterior for B depends only on D.

Scale invariance can be built into the model with the same procedure as described above for the simple (i.e. not translation invariant) WD-process. The posterior of α again follows an inverse Gamma distribution, and after introducing a prior with parameters (s_0, r_0) and integrating out α we arrive

at an expression analogous to (6.8) with $s = \frac{d}{4}\mathrm{tr}(\widetilde{W}D)$:

$$P(B|\bullet) \propto P_n(B|\xi,k)\det(\widetilde{W}_{(\alpha=1)})^{\frac{d}{2}}\,(s+s_0)^{-(n\frac{d}{2}+r_0)}\,. \tag{6.11}$$

6.1.4 Efficient Inference via Gibbs Sampling

In Gibbs sampling one iteratively samples parameter values from the full conditionals. Our model includes the following parameters: the partition B, the scale α, the covariance parameter θ, the number k of clusters in the population, the Dirichlet rate ξ and the degrees of freedom d. We propose to fix d, ξ and k: the **degrees of freedom** d might be estimated by the rank of S, which is often known from a pre-processing procedure. Note that d is not a very critical parameter, since all likelihood contributions are basically raised to the power of d. Thus, d might be used as an annealing-type parameter for "freezing" a representative partition in the limit $d \to \infty$. Concerning the **number k of clusters in the population**, there are two possibilities. Either one assumes $k = \infty$, which results in the Ewens-process model, or one expects a finite k. Our framework is applicable to both scenarios. Estimation of k, however is nontrivial if no precise knowledge about ξ is available. Unfortunately, this is usually the case, and $k = \infty$ might be a plausible assumption in many applications. Alternatively, one might fix k to a large constant which serves as an upper bound of the expected number, which can be viewed as truncating the Ewens process. The **Dirichlet rate** ξ is difficult to estimate, since it only weakly influences the likelihood. Consistent ML-estimators only exist for $k = \infty$ with $\hat{\xi} = k_B/\log n$, and even in this case the variance only decays like $1/\log(n)$, cf. [Ewen 72]. In practice, we should not expect to be able to reliably estimate ξ. Rather, we should have some intuition about ξ, maybe guided by the observation that under the Ewens process model the probability of two objects belonging to the same cluster is $1/(1+\xi)$. We can then either define an appropriate prior distribution, or we can fix ξ. Due to the weak effect of ξ on conditionals, these approaches are usually very similar.

The scale α can be integrated out analytically (see above). The distribution of θ is not of recognized form, and we propose to use a discretized prior set $\{p(\theta_j)\}_{j=1}^J$ for which we compute the posteriors $\{p(\theta_j|\bullet)\}_{j=1}^J$. A new value of θ is then sampled from the categorical distribution defined by $\{p(\theta_j|\bullet)\}_{j=1}^J$. In our implementation we use a uniform prior set ranging from $p(\theta_j) = 2/d$ to $p(\theta_{100}) = 200/d$. We define a *sweep* of the Gibbs sampler as one complete update of (B,θ). The most time consuming part in a sweep is the update

of B by re-estimating the assignments to blocks for a single object (characterized by a row/column in D), given the partition of the remaining objects. Therefore we have to compute the membership probabilities in all existing blocks (and in a new block) by evaluating equation (6.11), which looks formally similar to (6.8), but a factorization over blocks is no longer obvious. Every time a new partition structure is analyzed, a naive implementation requires $O(n^3)$ costs for computing the determinant of $\widetilde{W}$ and the product $\widetilde{W}D$. In one sweep of the sampler we need to compute k_B such probabilities for n objects, summing up to costs of order of $O(n^4 k_B)$.

Theorem 6.1.1 *Assuming k_B blocks in the actual partition and a fixed maximum iteration number in numerical root-finding, a sweep of the Gibbs sampler for the translation-invariant WD model can be computed in $O(n^2 + nk_B^2)$ time.*

Proof: Assume we want to compute the membership probabilities of the l-th object, given the partition of the remaining objects and all other parameter values. We first have to downdate all quantities which depend on object l and the block to which it is currently assigned, assign it to each of the existing blocks (and to a new block), and compute the probabilities of these events. With "downdate" we denote the reverse procedure to "update", i.e. we revert an assignment. From the resulting categorical distribution we then sample a new assignment (say block c) and update all quantities depending on object l and block c. We repeat this procedure for all objects $l = 1, \ldots, n$. Since downdating and updating are reverse to each other but otherwise identical operations, it suffices to consider the updating situation in which a new object with index l has to be assigned to a block in a given matrix B, or to a new block. To compute the membership probabilities we have to assign the new object to a block and evaluate (6.11) for the augmented matrix D_*, which has one additional column and row. For notational simplicity we will drop the subscript $_*$, since we will always consider the augmented quantities. Eq. (6.11) has two components: the prior $P(B|\xi, k)$ and the likelihood term which requires us to compute $\det(\widetilde{W}_{(\alpha=1)})$ and $\mathrm{tr}(\widetilde{W}D)$. Using the identity $\Gamma(x+1) = x\Gamma(x)$ in (5.1), the contribution of the prior is $n_c + \xi/k$ for existing clusters and $\xi(1 - k_B/k)$ for a new cluster (one simply sets $k = \infty$ for the Ewens-process).

For the likelihood term, consider first the generalized determinant $\det(\widetilde{W})$ in (6.11). Since $\widetilde{W} = W - (\mathbf{1}^T W \mathbf{1})^{-1} W \mathbf{1}\mathbf{1}^T W$, we have to compute $\rho := (\mathbf{1}^T W \mathbf{1})^{-1}$ for the augmented matrix W after assigning the new object l to block c. Analyzing (6.3) one derives $\rho^{-1} = \sum_{b \in B} n_b \lambda_b$, where $\lambda_b = (1 + \theta n_b)^{-1}$ are the k_B smallest eigenvalues of $W_{(\alpha=1)}$, see eq. (6.5).

Thus, we increase n_c, recompute λ_c and update ρ. Given ρ, we need to compute the eigenvalues of $W - \rho W\mathbf{1}\mathbf{1}^T W =: W - \rho \boldsymbol{v}\boldsymbol{v}^T$, where the latter term defines a rank-one update of W. Analyzing the characteristic polynomial, it is easily seen that the (size-ordered) eigenvalues $\tilde{\lambda}_i$ of $\widetilde{W}$ fulfill three conditions, see [Golu 89]: (i) the smallest eigenvalue is zero: $\tilde{\lambda}_1 = 0$; (ii) the largest $n - k_B$ eigenvalues are identical to their counterparts in W: $\tilde{\lambda}_i = \lambda_i,\ i = k_B + 1, \ldots, n$; (iii) for the remaining eigenvalues with indices $\tilde{\lambda}_2, \ldots, \tilde{\lambda}_{k_B}$ it holds that if λ_i is a repeated eigenvalue of W, $\tilde{\lambda}_i = \lambda_i$. Otherwise, they are the simple roots of the *secular* equation $f(y) = \rho + \sum_{j=1}^{k_B} \frac{n_j \lambda_j^2}{y - \lambda_j}$ fulfilling the relations $\lambda_i < \tilde{\lambda}_{i+1} < \lambda_{i+1}$. Note that f can be evaluated in $O(k_B)$ time, and with a fixed maximum number of iterations in the root-finding procedure, $\det(\widetilde{W})$ can be computed in $O(k_B)$. A sweep involves n "new" objects and k_B blocks. Thus, the costs sum up to $O(nk_B^2)$, summarized in Algorithm 4.

Algorithm 4: Cost for computing the likelihood in one sweep

```
for i = 1 to n do
    for c = 1 to k_B do
        n_c ← n_c + 1, recompute λ_c and update ρ ⇝ O(1)
        Find roots of secular equation ⇝ O(k_B)
```

For the trace $\mathrm{tr}(\widetilde{W}D)$ we have to compute

$$\begin{aligned} \mathrm{tr}(\widetilde{W}D) &= \mathrm{tr}(WD) - \rho \cdot \mathrm{tr}(W\mathbf{1}\mathbf{1}^T WD) \\ &= \mathrm{tr}(WD) - \rho \cdot \mathbf{1}^T WDW\mathbf{1}. \end{aligned} \tag{6.12}$$

We first precompute $\forall a \in B$: $\bar{D}_{ia} = \sum_{j \in a} D_{ij}$, which induces $O(n)$ costs since there are n summations in total. The first term in (6.12) is $\mathrm{tr}(WD) = \sum_{b \in B} \mathrm{tr}(D_{bb}) - \frac{\theta}{1+n_b\theta}\bar{D}_{bb}$, so we first update $\bar{D}$ by recomputing its c-th row/column: update $\gamma_c = n_c\lambda_c$ and $\forall a \in B : \bar{D}_{ac} \leftarrow \bar{D}_{ac} + \bar{D}_{ia} + D_{ii}\delta_{a,c} \rightsquigarrow O(k_B)$ time, and update the c-th summand in $\mathrm{tr}(WD)$ in constant time. Defining $\bar{D}_{ab} := \mathbf{1}_a^T D_{ab}\mathbf{1}_b$ and $\gamma_a := \frac{n_a\theta}{1+n_a\theta}$, the second term in (6.12) reads

$$\begin{aligned} \rho \sum_{ab \in B} \mathbf{1}_a^T W_a D_{ab} W_b \mathbf{1}_b &=: \rho \sum_{ab \in B} \overline{\Phi}_{ab}, \\ \overline{\Phi}_{ab} &= \bar{D}_{ab} - \gamma_a \bar{D}_{ab} - \gamma_b \bar{D}_{ab} + \gamma_a\gamma_b \bar{D}_{ab}. \end{aligned} \tag{6.13}$$

Since we have already updated γ and $\bar{D}$, it requires $O(k_B)$ time to update the c-th row.

Algorithm 5: Cost for computing the trace in one sweep

for $i = 1$ *to* n **do**
  $\forall a \in B$: $\bar{D}_{ia} = \sum_{j \in a} D_{ij} \rightsquigarrow O(n)$
  for $c = 1$ *to* k_B **do**
    Update $\bar{D} \rightsquigarrow O(k_B)$ Recompute c-th summand in $\mathrm{tr}(WD) \rightsquigarrow O(1)$ Compute $\forall a \in B : \overline{\Phi}_{ac} = \overline{\Phi}_{ca} \rightsquigarrow O(k_B)$

In a sweep, the costs for the trace sum up to $O(n^2+nk_B^2)$, see Algorithm 5.

The sweep is completed by resampling θ from a discrete set with J levels which induces costs of $O(k_B^2)$. Computing the discrete posterior involves J evaluations of both the determinant and the trace □

From the above theorem it follows that the worst case complexity in one sweep is $O(n^3)$ in the infinite mixture (i.e. Ewens process-) model, since $k_B \leq n$, and $O(n^2)$ for the truncated Dirichlet process with $k_B \leq k < \infty$. If the "true" k is finite, but one still uses the infinite model, it is very unlikely to observe the worst-case $O(n^3)$ behavior in practice: if the sampler is initialized with a one-block partition (i.e. $k_B = 1$), the trace of k_B typically shows an "almost monotone" increase during burn-in, see Figure 6.6 in the experiments section.

One possible extension of the TIWD cluster process is to include some **preprocessing step**. From the model assumptions $S \sim \mathcal{W}(\Sigma_B)$ it follows that if Σ_B contains k_B blocks and if the separation between the clusters (i.e. θ) is not too small, there will be only k_B dominating eigenvalues in S. Thus, one might safely apply kernel PCA to the centered matrix $S_c = -(1/2)QDQ$, i.e. compute $S_c = V\Lambda V^T$, consider only the first $\tilde{k}$ "large" eigenvalues in Λ for computing a low-rank approximation $\tilde{S}_c = V\tilde{\Lambda}V^T$, and switch back to dissimilarities via $\tilde{D}_{ij} = (\tilde{S}_c)_{ii} + (\tilde{S}_c)_{jj} - 2(\tilde{S}_c)_{ij}$. Such preprocessing might be particularly helpful in cases where $S_c = -(1/2)QDQ$ contains some negative eigenvalues which are of relatively small magnitude. Then, the low-rank approximation might be positive semi-definite so that $\tilde{D}$ contains squared Euclidean distances. Such situations occur frequently if the dissimilarities stem from pairwise comparison procedures which can be interpreted as approximations to models which are guaranteed to produce Mercer kernels. A popular example are classical string alignments which might be viewed as approximations of probabilistic alignments using pairwise hidden Markov models. We present such an example in Section 6.1.5. The downside of kernel PCA are the added costs of $O(n^3)$, but randomized approximation methods

have been introduced which significantly reduce these costs. In our TIWD software we have implemented a "symmetrized" version of the random projection algorithm for low-rank matrix approximation proposed in [Vemp 04] which uses the idea proposed in [Bela 07].

Another extension of the model concerns **semi-supervised** situations where for a subset of n_m observations class labels, i.e. assignments to k_m groups, are known. We denote this subset by the set of row indices $\mathbb{A} = \{1, \ldots, n_m\}$. Traditional semi-supervised learning methods assume that at least one labeled object per class is observed, i.e. that the number of classes is known. This assumption, however, is questionable in many real world examples. We overcome this limitation by simply fixing the assignment to blocks for objects in $\mathbb{A}$ during the sampling procedure, and re-estimating only the assignments for the unlabeled objects in $\mathbb{B} = \{n_m + 1, \ldots, n\}$. Using an Ewens process model with $k = \infty$ (or a truncated version thereof with $k = k' > k_m$), the model has the freedom to introduce new classes if some objects do not resemble any labeled observation. We present such an example below, where we consider protein sequences with experimentally confirmed labels (the "true" labels) and others with only machine predicted labels (which we treat as unlabeled objects).

6.1.5 Experiments

In a first experiment we compare the proposed TIWD cluster process with several hierarchical clustering methods on synthetic data, generated as follows: (i) a random block-partition matrix B of size $n = 500$ is sampled with $k_B = 10$; (ii) $d = 100$ samples from $\mathcal{N}(\mathbf{0}_n, \Sigma)$ are drawn, and arranged as the columns of the matrix $X_{(n \times d)}$, with $\Sigma = \alpha I_n + \alpha\theta B$, $\alpha = 2$ and different θ-values; (iii) squared Euclidean distances are stored in the matrix $D_{(n \times n)}$; (iv) this procedure is repeated 20 times.

A two-dimensional kernel PCA projection of an example distance matrix is shown in the left panels of Fig. 6.5 (large $\theta \leftrightarrow$ clear cluster separation in the upper panel, and small $\theta \leftrightarrow$ highly overlapping clusters in the lower panel). 5000 Gibbs sweeps are computed for the TIWD cluster process (after a burn-in phase of 2000 sweeps), followed by an annealing procedure to "freeze" a certain partition, cf. Section 6.1.4. For comparing the performance, several hierarchical clustering methods are applied: "Wards", "complete linkage", "single linkage", "average linkage", (see [Jain 88]), and the resulting trees are cut at the same number of clusters as found by TIWD. The right panels show the agreement of the inferred partitions with the true labels, measured

in terms of the adjusted rand index. If the clusters are well-separated, all methods perform very well, but for highly overlapping clusters, TIWD shows significant advantages over the hierarchical methods.

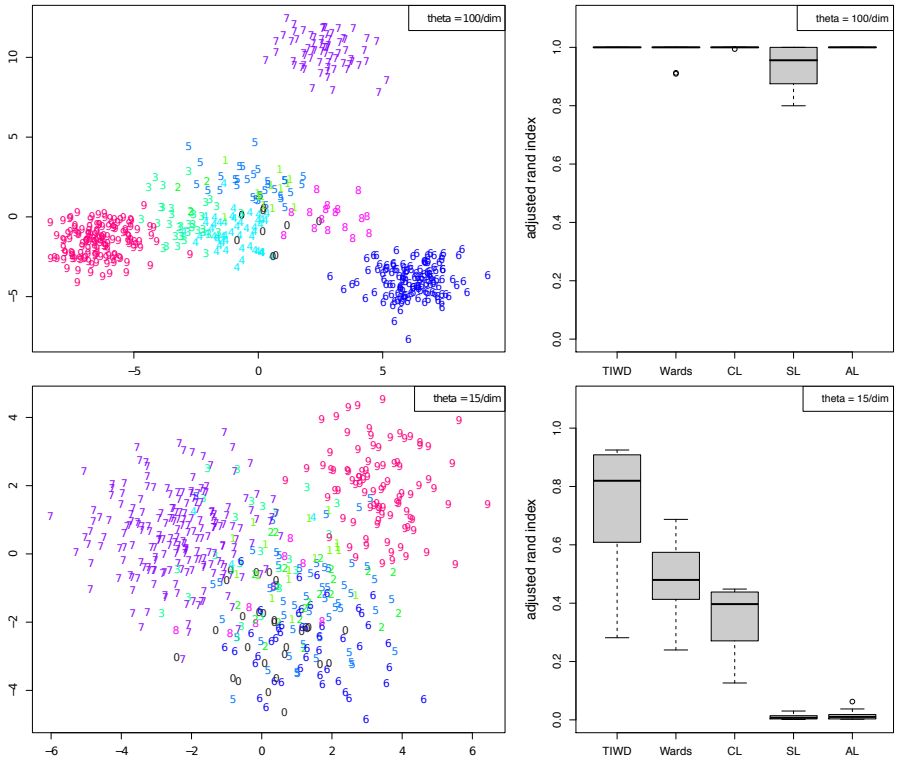

Figure 6.5: TIWD vs. hierarchical clustering ("Wards", "complete linkage", "single linkage", "average linkage") on synthetic data ($k = 10$, $n = 500$, $d = 100$, repeated 20 times).

In a second experiment we investigate the scalability of the algorithm to large data sets. The "small θ"-experiment above (lower panels in Fig. 6.5) is repeated for a large D-matrix of size (8000×8000). Figure 6.6 depicts the trace of the number of blocks k_B during sampling. The sampler stabilizes after roughly 500 sweeps. Note the remarkable stability of the sampler (compared to the usual situations in "traditional" mixture models), which follows from the fact that no label-switching can appear in the TIWD sampling algorithm. On a standard computer, this experiment took roughly two hours, which leads us to the conclusion that the proposed sampling algorithm is so efficient (at least for moderate k) that memory constraints are probably more severe than time constraints on standard hardware.

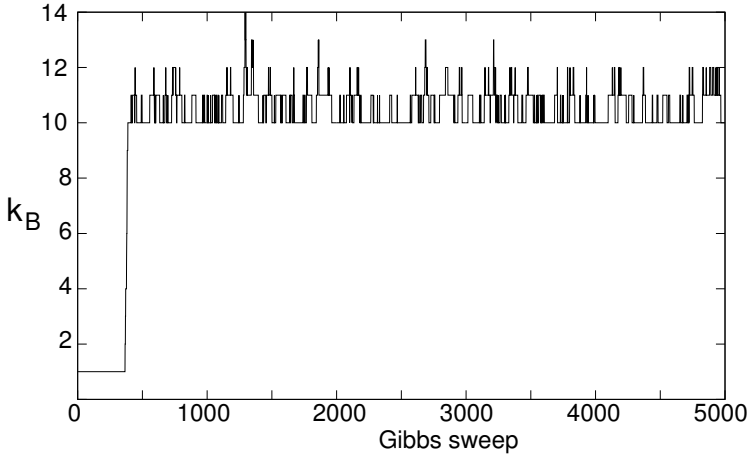

Figure 6.6: Trace-plot of the number of blocks k_B during the Gibbs sweeps for a large synthetic dataset. (10 clusters, $n = 8000$).

In a next experiment we analyze the influence of encoding the translation invariance into the likelihood (our TIWD model) versus the un-normalized WD process and row-mean normalization as described in Section 6.1.2. A similar random procedure for generating distance matrices is used, but this time we vary the number of replications d and the mean vector $\boldsymbol{\mu}$. If $\boldsymbol{\mu} = \mathbf{0}_n$, both the simple WD process and the TIWD process are expected to perform well, which is confirmed in the 1st and 3rd panel (left and right boxplots). Row-mean subtraction, however, introduces significant bias and variance. For nonzero mean vectors (2nd and 4th panel), the un-normalized process completely fails to detect the cluster structure, and row-mean subtraction can only partially overcome this problem. The TIWD process clearly outperforms the other models.

In a last experiment we consider a semi-supervised application example. In this experiment we study all globin-like protein sequences from the UniProtKB database with experimentally confirmed annotations and the TrEMBL database with unconfirmed annotations [UniP 10]. The former set consists of 1168 sequences which fall into 114 classes. These sequences form the "supervised" subset, and their assignments to blocks in the partition matrix are "clamped" in the Gibbs sampler. The latter set contains 2603 sequences which are treated as the "unlabeled" observations. Pairwise local string alignment scores s_{ij} are computed between all sequences and transformed into dissimilarities using an exponential transform. The resulting dissimilarity matrix D is not guaranteed to be of negative type (and indeed, $-QDQ$ has some small negative eigenvalues). We overcome this prob-

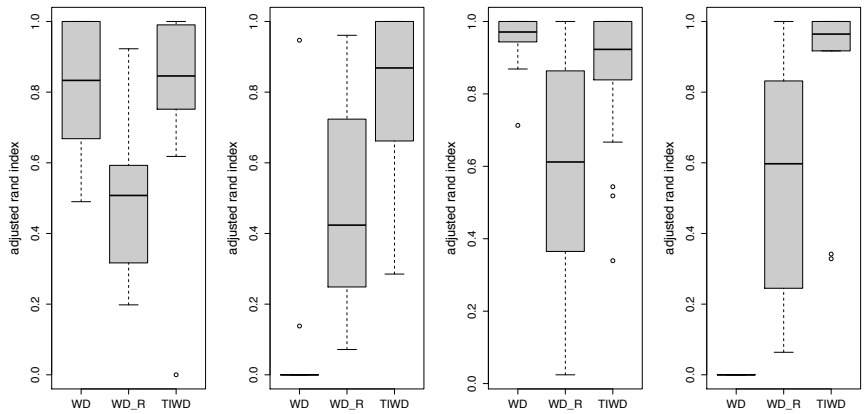

Figure 6.7: Comparison of WD and TIWD cluster process on synthetic data. "WD": WD without any normalization, "WD_R": WD with row mean subtraction. Left to right: (i) $d = 3, \boldsymbol{\mu} = \mathbf{0}$; (ii) $d = 3, \mu_i \sim N(40, 0.1)$; (iii) and (iv) same for $d = 100$.

lem by using the randomized low-rank approximation technique according to [Vemp 04, Bela 07], cf. Section 6.1.4, which effectively translates D into a matrix $\tilde{D}$ which is of negative type. The Ewens process model makes it possible to assign the unlabeled objects to existing classes or to newly created ones. Finally, almost all unlabeled objects are assigned to existing classes, with the exception of three new classes which have a nice biological interpretation. Two of the new classes contain globin-like bacterial sequences from *Actinomycetales*, a very special group of obligate aerobic bacteria which have to cope with oxidative stress. The latter might explain the existence of Redox domains in the globin sequences, like the Ferredoxin reductase-type (FAD)-binding domain observed in all sequences in one of the clusters and the additional nicotinamide adenine dinucleotide (NAD)-binding domain present in all sequences in the second new cluster, see Figure 6.8. Some of the latter sequences appear to be similar to another class that also contains *Actinomycetales* (see the large "off diagonal" probabilities surrounded by the blue circle) which, however, share a different pattern around some heme binding sites in the globin domain. The third newly formed class contains short sequence fragments which all show a certain variant of the Hemoglobin beta subunit. With the exception of the above mentioned similarity of one of the Actino-bacterial classes to another one, the three new classes show no similarity to any of the other classes, which nicely demonstrates the advantage of a semi-supervised learning model that is flexible enough to allow the creation of new groups.

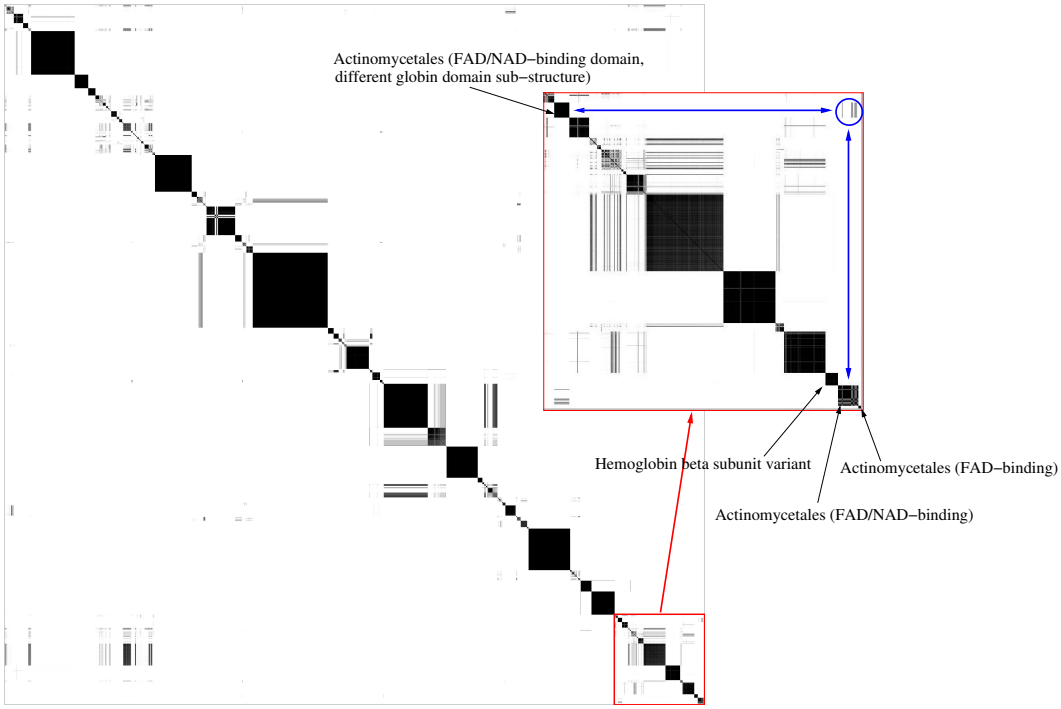

Figure 6.8: Co-membership probabilities of globin proteins. Three new classes which have a nice biological interpretation are detected. Two of the new classes contain globin-like bacterial sequences. The third new class contains short sequence fragments which all show a certain variant of the Hemoglobin beta subunit. All the three newly detected classes show no similarity to any of the other classes.

6.2 Multi-View Clustering of Distance Data

In this section we consider the problem of clustering multiple instances of pairwise distances D. We extend the single-view clustering model introduced in Section 6.1 to cluster different views of co-occurring samples. We think of each view as one realization of a distance matrix. The aim is to obtain a combined clustering of all views and benefit from the shared structural information in the different views.

The particular challenges arising here are the following: In the single-view model introduced in Section 6.1, the data was assumed to be distributed according to a normal distribution with covariance matrix $\Sigma_B = \alpha I_n + \beta B$ where α and β denoted a scalar value and B a block matrix. The geometric interpretation for such a covariance matrix is that all clusters have the same between-class variance, i.e. all clusters are equidistant. If we assumed this for the multi-view case, we would restrict the geometric cluster configurations to be identical across all views, which would be a serious limitation. Hence, for the multi-view clustering scenario, we want to encode more degrees of freedom to be able to differentiate between geometric cluster arrangements over different views. Therefore, the covariance matrix is chosen to be a full, symmetric block matrix, where every diagonal/upper diagonal block may have a separate β-value, allowing for maximum flexibility. In addition, a novel translation-invariant likelihood has to be chosen.

We introduced the concept of multi-view learning in vector spaces in Section 2.4.2. As our focus in this work lies on (dis)similarity data, we first generalize the vector-space approach step by step to inner-product spaces in Section 6.2.1 and then advance to incorporating invariances that are crucial for dealing with pairwise distances. In Section 6.2.2 we propose our new model for partitioning distance data that is available in multiple views. We call this model *Multi-View Translation-invariant Dirichlet (MVTID) Clustering Process*. More precisely, we aim at modeling dependencies between co-occurring data sets, i.e. we concentrate on a subfield of multi-view clustering, the so-called dependency-seeking clustering, as explained in Section 2.4.2. Finally, in Section 6.2.4 we present results of both synthetic and real world experiments.

6.2.1 Generalization of Vector Spaces to Inner-Product Spaces

Assume that the rows of the data matrix X are ordered according to cluster assignments, i. e. $X \sim \mathcal{N}(M, I_n \otimes \Gamma)$ for $X \in \mathbb{R}^{n \times d}$ and covariance matrix $\Gamma \in \mathbb{R}^{d \times d}$ with mean matrix $M \in \mathbb{R}^{n \times d}$ that has cluster-specific block structure.

For non-zero mean, $X\Gamma^{-1}X^T$ is distributed according to a non-central Wishart distribution, which causes severe computational problems due to the appearance of the hypergeometric function (c.f. [Gupt 00]).

However, we are able to approximate the non-central Wishart by a central Wishart distribution, yielding

$$X\Gamma^{-1}X^T \sim \mathcal{W}_d(\underbrace{\tfrac{1}{n}MM^T + I_n}_{=:\Sigma}). \tag{6.14}$$

which corresponds to $X \sim \mathcal{N}(0, \Sigma \otimes I)$. By using this approximation, the first order moments of the Wishart and the noncentral Wishart distribution are identical, whereas the second order moments differ in terms of order $\mathcal{O}(n^{-1})$. See [Gupt 00] for more detailed information.

Given $X \sim \mathcal{N}(0, \Sigma \otimes \Gamma)$, [McCu 08a] states that the log likelihood in its most general form is written as

$$\begin{aligned} l(\Sigma, \Gamma; X) &= -\tfrac{1}{2}\log\det(\Sigma \otimes \Gamma) - \tfrac{1}{2}\operatorname{tr}(X^T\Sigma^{-1}X\Gamma^{-1}) \\ &= -\tfrac{d}{2}\log|\Sigma| - \tfrac{n}{2}\log|\Gamma| - \tfrac{1}{2}\operatorname{tr}(X^T\Sigma^{-1}X\Gamma^{-1}). \end{aligned} \tag{6.15}$$

As in the Sections above, we again use the symbol $W := \Sigma^{-1}$ for convenience. In order to see that (6.15) is formulated in the inner-product space, it suffices to apply cyclic permutation inside the trace arriving at the term $WX\Gamma^{-1}X^T$.

For the choice of $\Gamma = I_d$, i.e. $X\Gamma^{-1}X^T = XX^T =: S$, we arrive at the central Wishart model like in Section 6.1. However, spherical covariances are an extreme case: all dimensions within one view are treated separately, meaning we cannot distinguish between dimensions and views.

Up to this point, we formulated the model for inner-product spaces. But, as we only observe pairwise distances, we cannot recover any information about scaling and the origin of X, i.e. translations or column shifts. This is why the model is further required to be invariant against transformations of such kind.

Now we assume a general, positive-definite Γ. For a fixed covariance matrix Σ, the log likelihood is maximized at $\hat{\Gamma}_\Sigma = \frac{1}{n} X^T W X$. Hence, the profile log likelihood l_p is written as the following (see [McCu 08a], Model III):

$$l_p(\Sigma; X) = -\tfrac{d}{2} \log |\Sigma| - \tfrac{n}{2} \log |X^T W X|. \tag{6.16}$$

We refer to this as Model A. It was shown in [McCu 08a] that in this case, the profile likelihood (6.16) is a "true" likelihood.

It is important to stress that the likelihood (6.16) is only informative if $d < n$: for $d = n$ the determinant $|X^T W X| = |W X X^T|$ splits into $|W| \cdot |X X^T|$, which completely removes Σ from the likelihood.

The assumption of a general, positive-definite Γ leads to the product-space setting introduced in (2.8). In order to perform dependency-seeking clustering, we now impose a constraint on Model A, where Γ is positive definite, but additionally has a block diagonal structure:

$$\Gamma = \begin{pmatrix} \Gamma_1 & 0 & \cdots & 0 \\ 0 & \Gamma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Gamma_T \end{pmatrix}. \tag{6.17}$$

Here, each Γ_t, $t \in \{1, \ldots, T\}$, is a positive-definite matrix of arbitrary size $d_t \times d_t$, where $\sum_{t=1}^{T} d_t = d$. By using such a block diagonal Γ matrix, the log likelihood (6.16) can be split into T terms:

$$l(\Sigma; X) = \sum_{t=1}^{T} -\tfrac{d_t}{2} \log |\Sigma_t| - \tfrac{n}{2} \log |X_t^T W_t X_t|. \tag{6.18}$$

We call this Model B. Note that this model allows T different data sets X_t and Σ_t, which leads to a new interpretation: We may imagine T different data sets observing the same n objects, but originating from different sources or methods of measurement. This is exactly what was previously introduced as *view* and, in accordance to (2.9), the model is dependency-seeking.

Translation Invariance

The step from Model A to Model B is necessary to capture dependencies between views. Still, operating only on pairwise distances D leads to the problem of not being able to recover any translation. Therefore, analog to

Section 6.1, the likelihood is altered to be invariant against arbitrary column shifts of the data. [McCu 08a] showed that if there is a whitened, shift-invariant data matrix $X = X - \mathbf{1}\boldsymbol{\mu}^T$ with some shift vector $\boldsymbol{\mu}$, then the log likelihood of Model A is given by

$$\check{l}(\Sigma; X) = \tfrac{d}{2} \log\det(WQ) - \tfrac{n-1}{2} \log\det(X^T WQX). \tag{6.19}$$

$Q = I_n - \mathbf{1}_n(\mathbf{1}_n^T W \mathbf{1}_n)^{-1}\mathbf{1}_n^T W$ denotes a projection matrix and $\det(\cdot)$ is the product of non-zero eigenvalues. Note that equation (6.19) can also be formulated in terms of distances D where $D_{ij} = S_{ii} + S_{jj} - 2S_{ij}$:

$$\check{l}(\Sigma; D) = \tfrac{d}{2} \log\det(WQ) - \tfrac{n-1}{2} \log\det(-\tfrac{1}{2}WQD). \tag{6.20}$$

For Model B, this yields

$$\check{l}(\Sigma; X) = \sum_{t=1}^{T} \check{l}(\Sigma_t; X_t) \tag{6.21}$$

$$= \sum_{t=1}^{T} \tfrac{d_t}{2} \log\det(W_t Q_t) - \tfrac{n-1}{2} \log\det(X_t^T W_t Q_t X_t). \tag{6.22}$$

With $Q_t = I_n - \mathbf{1}_n(\mathbf{1}_n^T W_t \mathbf{1}_n)^{-1}\mathbf{1}_n^T W_t$ there exists a separate projection matrix for each view. Again, this likelihood can be written in terms of D:

$$\check{l}(\Sigma; D) = \sum_{t=1}^{T} \tfrac{d_t}{2} \log\det(W_t Q_t) - \tfrac{n-1}{2} \log\det(-\tfrac{1}{2}W_t Q_t D_t) \tag{6.23}$$

With these theoretical results we are now able to extend the original TIWD model by using the likelihood introduced above to a dependency-seeking clustering approach. We call this model the *Multi-View translation-invariant Dirichlet clustering process* for clustering distance data.

6.2.2 The Multi-View Clustering Process

The assumption in this model is that the data is available in T different views. X_t, S_t and D_t denote the corresponding data-, similarity- and distance- matrices for a view t with $t \in \{1, ..., T\}$. In the TIWD clustering process introduced in Section 6.1, the columns of X were independent n-dimensional vectors distributed according to a normal distribution with covariance matrix $\Sigma_B = \alpha I_n + \beta B$ where β denoted a scalar value. The geometric interpretation

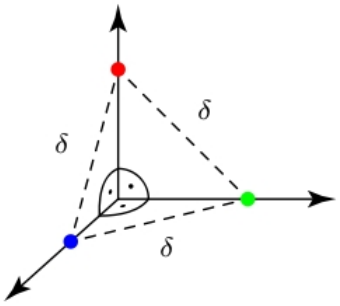

Figure 6.9: Example for three blocks with spherical between-class covariance matrix and a scalar value for β: cluster centers need to have the same distances δ.

for such a covariance matrix is that all clusters have the same between-class variance, i.e. all clusters are equidistant. This scenario is illustrated in Figure 6.9.

This means, using a covariance matrix $\Sigma_B = \alpha I_n + \beta B$ for the multi-view case, we would restrict the geometric cluster configurations to be identical across all views, which would be a serious limitation. Hence, for the multi-view clustering scenario, we want to encode more degrees of freedom that enable to differentiate between geometric cluster distances over different views. Therefore, the between-class covariance matrix $\Sigma = MM^T + I_n$ is chosen to be a full, symmetric $n \times n$ block matrix, allowing for arbitrary geometric cluster configurations. A graphical example for this scenario is depicted in Figure 6.10.

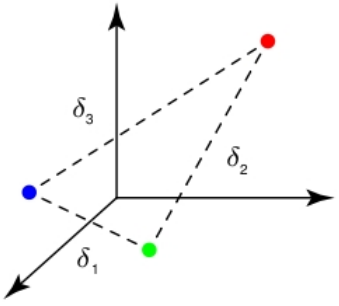

Figure 6.10: Exemplary cluster configuration with a between-class covariance matrix in full block form: all clusters are allowed to have different distances to each other.

For formulating a construction principle, we propose an intermediate step, namely introducing a smaller matrix K_t of size $k_B \times k_B$ that stores only one single β value per block. Having distinct β values, we now expand K_t into covariance matrices $\left(MM^T\right)_t \in \mathbb{R}^{n\times n}$ by duplicating elements according to the block sizes defined in the partition matrix B. The scheme can most easily be explained by the following example:

Assume $k_B = 3$ blocks, $n_1 = 2$, $n_2 = 2$ and $n_3 = 1$. Then, B is a block-diagonal matrix with 3 blocks of ones on the diagonal. $\left(MM^T\right)_t$ is received by filling the first diagonal block of B with $\beta_{t_{11}}$, the second with $\beta_{t_{22}}$, the third with $\beta_{t_{33}}$ and the off-diagonals with corresponding $\beta_{t_{ij}}$:

$$B = \left(\begin{array}{cc|cc|c} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 \end{array}\right), \qquad K_t = \begin{pmatrix} \beta_{t_{11}} & \beta_{t_{12}} & \beta_{t_{13}} \\ \beta_{t_{12}} & \beta_{t_{22}} & \beta_{t_{23}} \\ \beta_{t_{13}} & \beta_{t_{23}} & \beta_{t_{33}} \end{pmatrix}$$

$$\Longrightarrow \qquad \left(MM^T\right)_t = \left(\begin{array}{cc|cc|c} \beta_{t_{11}} & \beta_{t_{11}} & \beta_{t_{12}} & \beta_{t_{12}} & \beta_{t_{13}} \\ \beta_{t_{11}} & \beta_{t_{11}} & \beta_{t_{12}} & \beta_{t_{12}} & \beta_{t_{13}} \\ \hline \beta_{t_{12}} & \beta_{t_{12}} & \beta_{t_{22}} & \beta_{t_{22}} & \beta_{t_{23}} \\ \beta_{t_{12}} & \beta_{t_{12}} & \beta_{t_{22}} & \beta_{t_{22}} & \beta_{t_{23}} \\ \hline \beta_{t_{13}} & \beta_{t_{13}} & \beta_{t_{23}} & \beta_{t_{23}} & \beta_{t_{33}} \end{array}\right)$$

In general, the symmetric block matrix $\left(MM^T\right)_t$ can also be computed with the help of a matrix Z:

$$Z = \begin{pmatrix} \mathbf{1}_{n_1} & \mathbf{0}_{n_1} & \cdots & \mathbf{0}_{n_1} \\ \mathbf{0}_{n_2} & \mathbf{1}_{n_2} & \cdots & \mathbf{0}_{n_2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{n_b} & \mathbf{0}_{n_b} & \cdots & \mathbf{1}_{n_b} \end{pmatrix} \in \mathbb{R}^{(n\times k_B)},$$

where $\mathbf{1}_{n_b}$ denotes a n_b-vector of ones and $\mathbf{0}_{n_b}$ a n_b-vector of zeros. Using this, we have

$$\left(MM^T\right)_t = ZK_tZ^T. \tag{6.24}$$

Thus, the columns of the data matrix X_t we consider in the MVTID clustering process are n-dimensional vectors $\boldsymbol{x}_i \in \mathbb{R}^n$, $i = 1, \ldots, d$, distributed according to a zero-mean Gaussian with covariance matrix $\Sigma_t = \alpha I_n + \left(MM^T\right)_t$.

Despite the additional degrees of freedom given by Σ_t, it can not account for the problem of translation invariance, meaning, we still need a likelihood

that is constant over all similarity matrices S. The reason for this can be seen by going back to the definition of squared pairwise distances $D \in \mathbb{R}^{n \times n}$

$$D_{ij} = S_{ii} + S_{jj} - 2S_{ij}. \tag{6.25}$$

As explained in Section 6.1, a distance matrix D does not carry any information about the origin of the coordinate system anymore. As a consequence, going in reverse and constructing S does not yield just one matrix, but a whole equivalence class

$$\mathbb{S} := \left\{ S | S = \tilde{S} + \mathbf{1}\boldsymbol{v}^T + \boldsymbol{v}\mathbf{1}^T, S \succeq 0, \boldsymbol{v} \in \mathbb{R}^n \right\} \tag{6.26}$$

that maps to D. Here, $\boldsymbol{v}$ is a vector of n unknown parameters, effectively shifting all columns of X_t. If all parameters v_i, $i = 1 \dots n$, have different values, the resulting S completely loses the block structure of $\tilde{S}$. Hence, even a full block matrix Σ_t on its own can not infer the exact form of S. We need a model that is independent under column shifts in X, which is why we encode the translation invariance directly into the likelihood.

6.2.3 Efficient Inference via Gibbs sampling

As mentioned above, all views are assumed to be independent given a partition B, hence the likelihood for B and K_t factorizes for all views:

$$\begin{aligned} &p(B, K_1, \dots, K_t | X_1, \dots, X_t, \bullet) \\ &\propto \prod_{t=1}^{T} \exp\left(\check{l}(\Sigma_t; X_t)\right) \; P(K_t)\, P(B|\xi, k), \end{aligned} \tag{6.27}$$

In order to compute the posterior, we propose to apply Gibbs sampling. Consider the following conditional distribution at view t

$$p(B, K_t | X_t, \bullet) \propto \exp\left(\check{l}(\Sigma_t; X_t)\right) \; P(K_t)\, P(B|\xi, k). \tag{6.28}$$

$P(K_t)$ is given by a Wishart distribution and updated via a Metropolis-Hastings sampling in every iteration of the Gibbs sampler.

As in the original TIWD model, the prior for block matrix B is defined to be Dirichlet-Multinomial over partitions, see equation 5.1. Algorithm 6 explains

the full sampling scheme in detail.

Algorithm 6: Gibbs sampler for multi-view clustering.

A : **Initialize** set $K_1 = sI_{n_1}$, $s > 0$, $k_B = 1$.
for $i = 1$ *to iteration* **do**
  for $j = 1$ *to* n **do**
    for $k = 1$ *to* k_B **do**
      assign object j to an existing cluster k or a new one
      update k_B
  for $t = 1$ *to* T **do**
    sample new K_t matrix using Metropolis Hastings
  compute likelihood (6.27)

Metropolis-Hastings Update Step. In the Metropolis-Hastings algorithm (see [Robe 05] for more details), a sequence of random samples is obtained from a probability distribution for which direct sampling is difficult. In Algorithm 6, at the end of every iteration of the Gibbs sampler, the matrix K_t is updated for every view. In the following explanation, we will skip the index t for simplicity and consider the Metropolis Hastings update for one view t, i.e. in this paragraph we define $K := K_t$. A new matrix denoted by K_p is proposed via the *proposal* distribution $q(K_p|K_{old})$ with $K_p \sim \mathcal{W}(K_{old})$ where K_{old} denotes the current K matrix before the update. Using the conditional density q, a Markov chain is produced as shown in Algorithm 7.

Algorithm 7: Metropolis-Hastings

Given K_{old}.
Take

$$K_{new} = \begin{cases} K_p & \text{if} \quad \text{Unif}(0,1) \leq p(K_{old}, K_p) \\ K_{old} & \text{otherwise} \end{cases}$$

where $p(K_{old}, K_p) = \min\left\{\frac{f(K_p)}{f(K_{old})}\frac{q(K_{old}|K_p)}{q(K_p|K_{old})}, 1\right\}$

Thereby, $p(K_{old}, K_p)$ is called Metropolis-Hastings *acceptance* probability and $f(K) := \exp(\check{l}(\Sigma; X))\mathcal{W}(K|I_{k_B})$.

It is important to note that we always have to construct a positive definite $\widetilde{K}$ matrix that consist of the current K matrix with one additional row and column to account for a new cluster, i. e.

$$\widetilde{K} = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix} \tag{6.29}$$

with $K_{11} = K \in \mathbb{R}^{k_B \times k_B}$, $K_{21} \in \mathbb{R}^{1 \times k_B}$, $K_{12} = K_{21}^T$ and $K_{22} \in \mathbb{R}$.

To ensure the positive definiteness of $\widetilde{K}$, the additional row and column are computed by the following (see [Bilo 99] for details):

$$K_{12}|K_{11} \sim \mathcal{N}(\mathbf{0}, K_{11} \otimes s) \tag{6.30}$$

$$K_{22.1} \sim \mathcal{W}_1(d - k_B, s) \tag{6.31}$$

$$K_{22} = K_{22.1} + K_{21}K_{11}^{-1}K_{12}. \tag{6.32}$$

Complexity Analysis of Model B For reasons of simplicity, we only analyze one view t throughout this section as well and thus drop the index from all view-dependent terms (X_t, d_t, S_t, Σ_t, W_t, Q_t and K_t). Since we know the total number of views beforehand, T is a constant factor and therefore disregarded.

In its simplest form, computing the likelihood for one Gibbs sweep consists of assigning all n objects to k_B existing blocks and 1 new block, each step involving the inverse of the full $n \times n$ covariance matrix Σ. In total, this adds up to a cost of $\mathcal{O}(k_B n^4)$, although, due to the block structure of Σ and W, we may employ several computational shortcuts that reduce the complexity.

To ensure the data does not violate the model constraint $d < n$, we first calculate X as a low-rank PCA projection of S in a $\mathcal{O}(n^3)$ pre-processing step. While this embedding is needed to ensure $d < n$, we may as well use the computational benefits of the likelihood in X, instead of recomputing S or D. Still, the likelihood is constructed to incorporate scale and translation invariance, which means that $\check{l}(\Sigma; X)$ is fully equivalent to $\check{l}(\Sigma; S)$ and $\check{l}(\Sigma; D)$.

Theorem 6.2.1 *Given X of size $n \times d$, the computational cost of one complete Gibbs sweep in Model B can be computed in $\mathcal{O}(nk_B d^3 + nk_B^4 + nk_B^3 d + nk_B^2 d^2)$ time.*

Proof: The translation-invariant likelihood (6.21) reads

$$\check{l}(\Sigma; X) = \tfrac{d}{2} \log\det(WQ) - \tfrac{n-1}{2} \log\det\left(X^T W Q X\right).$$

[McCu 09] showed that $\det(WQ)$ can be reformulated in terms of W

$$\det(WQ) = n(\mathbf{1}^T W \mathbf{1})^{-1} \det(W) \tag{6.33}$$

leading to

$$\begin{aligned}\check{l}(\Sigma; X) = \tfrac{d}{2} \log\left[n(\mathbf{1}^T W \mathbf{1})^{-1} \det(W)\right] - \\ \tfrac{n-1}{2} \log\det\left(X^T W X - (\mathbf{1}^T W \mathbf{1})^{-1} X^T W \mathbf{1}\mathbf{1}^T W X\right).\end{aligned}$$

As the covariance matrix has block structure, its inverse shares the exact same block structure, although with different values. Hence, W can be formulated as

$$W = ZLZ^T + \gamma I_n \tag{6.34}$$

with a a symmetric $k_B \times k_B$ matrix L and scaling factor γ. It holds:

$$I_n = \Sigma W = ZKZ^T ZLZ^T + \gamma ZKZ^T + \alpha ZLZ^T + \alpha\gamma I_n \tag{6.35}$$

Solving (6.35) yields: $L = -\frac{1}{\alpha}\left[KZ^T Z + \alpha I_{k_B}\right]^{-1} K$ and $\gamma = \frac{1}{\alpha}$.

Efficient updating scheme: During Gibbs sampling, we either move an object from one cluster to another or open a new one. All products involving Z can therefore be updated instead of fully recomputed. For instance, $Z^T Z$ is a diagonal $k_B \times k_B$ matrix, whose elements count the current number of objects per block. In the Gibbs sampler, we start with $k_B = 1$ and $Z^T Z = n$, which involves no cost at all.

Using an updating scheme, the computation of L consumes only $\mathcal{O}(k_B^3)$ due to a $k_B \times k_B$ matrix inversion. The computation of

$$\mathbf{1}^T W \mathbf{1} = \mathbf{1}^T ZLZ^T \mathbf{1} + \tfrac{1}{\alpha} n, \tag{6.36}$$

costs $\mathcal{O}(k_B^2)$, because $Z^T \mathbf{1}$ simply is the diagonal of $Z^T Z$. In order to find $\det(W) = \det(\Sigma)^{-1}$, we first decompose K into $K^{\frac{1}{2}} K^{\frac{1}{2}}$ in $\mathcal{O}(k_B^3)$ and write

$$ZKZ^T = ZK^{\frac{1}{2}} K^{\frac{1}{2}} Z^T, \tag{6.37}$$

which has the same non-zero eigenvalues as the $k_B \times k_B$ matrix $A := K^{\frac{1}{2}} Z^T Z K^{\frac{1}{2}}$. A singular value decomposition of $A = UCV^T$ in $\mathcal{O}(k_B^3)$ finally leads to

$$\det(\Sigma) = \alpha^{n-k_B} \prod_{i=1}^{k_B} C_{ii} + \alpha. \tag{6.38}$$

For the remaining terms in the likelihood, we have

$$X^T W X = X^T Z L Z^T X + \tfrac{1}{\alpha} X^T X \tag{6.39}$$

and

$$X^T W \mathbf{1} = X^T Z L Z^T \mathbf{1} + \tfrac{1}{\alpha} X^T \mathbf{1}. \tag{6.40}$$

Here, $X^T Z$ is a $d \times k_B$ matrix, where each column $j \in \{1, \ldots, k_B\}$ is the sum of all (X^T)-columns $i \in \{1, \ldots, n\}$ that Z assigns to block j. This means that switching one object from one block to another actually means subtracting its (X^T)-column from its currently assigned $(X^T Z)$-column and adding it to a different one. In a nutshell, updates of $X^T Z$ are computed with constant cost. Initially, all objects are assigned to one cluster, so $X^T Z = X^T \mathbf{1}$, which involves $\mathcal{O}(nd)$.

$X^T X$ does not change throughout the sampling process and is hence pre-computed in $\mathcal{O}(nd^2)$. Consequently, equation (6.39) and (6.40) are of cost $\mathcal{O}(k_B^2 d + d^2 k_B)$ and $\mathcal{O}(k_B^2 d)$, and $X^T W \mathbf{1}\mathbf{1}^T W X = \left(X^T W \mathbf{1}\right)\left(X^T W \mathbf{1}\right)^T$ takes $\mathcal{O}(k_B^2 d + d^2)$. The determinant of a $d \times d$ matrix is computed as the product of non-zero eigenvalues, involving $\mathcal{O}(d^3)$. In total, computing the full likelihood one single time has cost $\mathcal{O}(d^3 + k_B^3 + k_B^2 d + d^2 k_B)$. A complete sweep of the Gibbs sampler requires the likelihood to be calculated nk_B times, arriving at $\mathcal{O}(nk_B d^3 + nk_B^4 + nk_B^3 d + nk_B^2 d^2)$. □

Due to the low-rank PCA projection, d is a constant smaller than n, so we have a cost $\mathcal{O}(nk_B^4)$ for one Gibbs sweep. If we further use a truncated Dirichlet process, the complexity reduces to only $\mathcal{O}(n)$.

6.2.4 Experiments

Synthetic Experiment. In a first experiment, we test our method on synthetic data. Here, $T = 2$ views are considered, which both have $k_B = 3$ clusters of $n = 200$ objects in $d = d_1 = d_2 = 2$ dimensions. The data is

generated in the following way: A random $n \times (Td = 4)$ matrix A is sampled from a zero-mean multivariate Normal distribution with covariance matrix $K = I_{(Td)}$. We sort column 1 and 3 from low to high values and then divide them into k_B randomly-sized subsets. This effectively introduces coupling between both columns concerning low/medium/high values. Afterwards, each subset is permuted randomly to reverse the sorting effect. View 1 with $n \times d$ matrix X_1 is chosen to be column 1 and 2 of A, view 2 with $n \times d$ matrix X_2 consists of column 3 and 4. The rightmost plot of Figure 6.11 shows the explicitly constructed correlation between view 1 and 2 that is used for determining the true labels. The two remaining plots visualize the labeling applied to both views.

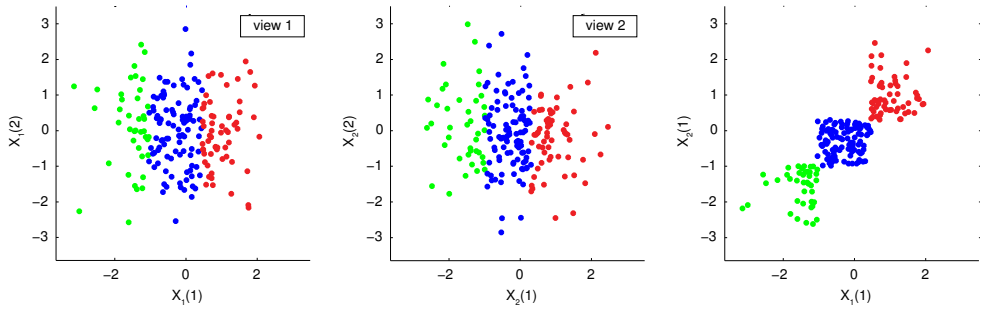

Figure 6.11: Exemplary synthetic dataset with $n = 200$ objects, $k_b = 3$ clusters in $T = 2$ views. The colors correspond to the true labeling. View 1 and 2 are constructed to show no (significant) correlation when seen individually (single view) or when combined into one data matrix (product space). Only the multi-view setting can dissolve the inter-view dependency structure of the rightmost plot and adjust the clustering accordingly.

This whole procedure is repeated 100 times to generate multiple data sets, each of which is subsequently clustered using i) only view 1, ii) only view 2, iii) the product space of view 1 and 2 and iv) view 1 and 2 jointly (multi-view). For all methods, we use the same implementation running with one identical set of parameters for 1000 Gibbs sweeps and then compute the adjusted Rand index to the true labeling of each data set. The final results can be seen in Figure 6.12. As expected, both single view clusterings only see non-correlated data and thus on average assign all objects to one big cluster, leading to an adjusted Rand index of zero. Clustering of the product space takes both views into account, however by discarding which dimensions come from which view. In that case, the views lose their semantic meaning and reduce to additional dimensions. Only if we enforce inter-view independency by writing

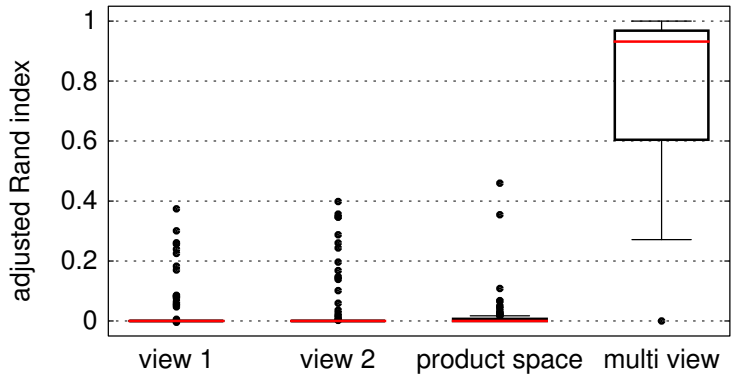

Figure 6.12: Boxplot of the adjusted Rand index between the true labels and clustering assignments, repeated 100 times, 1000 Gibbs sweeps each.

the likelihood as a sum of independent terms $l(\Sigma_t; X_t)$, we intentionally create a model mismatch. As a result, the model compensates this via adjusting the intra-view covariances, which eventually leads to introducing more clusters. This is why, in contrast to all previous methods, multi-view treatment of the data is the only approach to successfully recover the dependency structure.

Real World Experiment. In this experiment we focus on clustering a certain type of human proteins, namely the so-called *proteases*. Proteases are cellular enzymes that conduct proteolysis, i.e. the directed degradation (digestion) of proteins. Proteases are interesting from a medical viewpoint, since they play a key role in the development of metastatic tumors and in the reproductive cycle of certain viruses like HIV. Within the so-called Enzyme Commission number nomenclature, the proteases form the class 3.4, which again is further hierarchically subdivided into 14 subclasses. These subclasses are defined according to the type of catalyzed reaction and structural properties of the active center (which is the part of an enzyme where substrates bind and undergo a chemical reaction). It is well known that the class definition in the EC-system is problematic, since these classes do not take into account evolutionary relations between the enzymes. Such evolutionary relations, on the other hand, should be reflected in the similarity of the enzymes' amino acid sequences. Therefore, when it comes to detecting the underlying structure of proteases enzymes by way of clustering, it seems to be promising to use a multi-view approach where structural features form one view, and sequential features form a second view.

In our clustering experiment we collect all known protein structures of human

proteases in the PDB database[1] (view 1), as well as the corresponding amino acid sequences (view 2). To remove (near) duplicates we select the subset of 193 proteins with less than 95% sequence identity. In order to derive pairwise distances for the structures in view 1, we use an information-theoretic approach: Given two strings x and y and denoted by $K(\cdot)$ the Kolmogorov complexity, the *Normalized Information Distance* is defined as

$$NID(x,y) = \frac{K(xy) - \min\{K(x), K(y)\}}{\max\{K(x), K(y)\}}, \tag{6.41}$$

where $K(xy)$ is the binary length of the shortest program that produces the pair x, y. As a computable approximation it has been proposed in [Cili 05] to use the *Normalized Compression Distance:*

$$NCD(x,y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}, \tag{6.42}$$

where $C(xy)$ represents the size of the file obtained by compressing the concatenation of x and y. In our setting the strings x, y are vectorized *contact maps* computed from the protein structures. A schematic overview is shown in Figure 6.13.

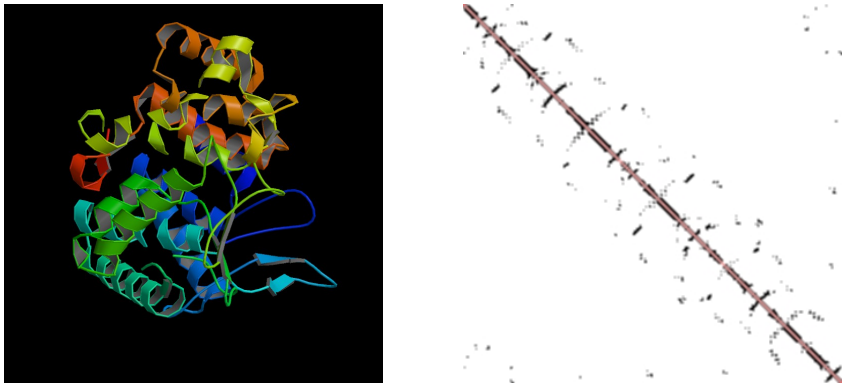

Figure 6.13: Computing pairwise distances between protein structures by first calculating binary contact maps. Contact maps are binary matrices in which the (i, j)-the element is 1 if two residues are closer than a predetermined threshold, and 0 otherwise. The contact maps are then transformed into binary strings (by column-wise vectorization), which finally are used to compute the compression distances using the *bzip2* text compressor.

[1] http://www.rcsb.org/pdb/home/home.do

For pairs of amino acid sequences (a_i, a_j) in view 2, we compute length-normalized string alignment scores $s_{ij}^{\text{norm}} = s_{ij} / \min\{l(a_i), l(a_j)\}$, where s_{ij} is the Smith-Waterman alignment score and $l(a_i)$ is the length of sequence a_i. These scores were transformed into pairwise distances according to $d_{ij} = \exp(-c \cdot s_{ij}^{\text{norm}})$. From the two distance matrices D_1 (view 1) and D_2 (view 2) we compute two representative matrices by using the centering transformation $S^c = -\frac{1}{2} D^c = -\frac{1}{2} Q_I D Q_I^T$. A low-rank approximation of each of these S matrices via kernel-PCA finally yields X_1 and X_2 with 20 columns each.

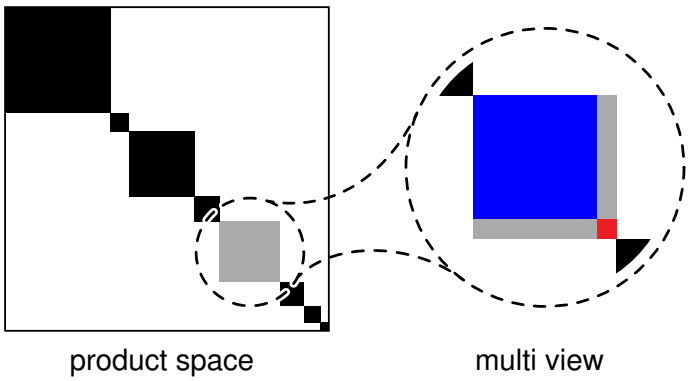

Figure 6.14: Clustering of protein sequences. The left box refers to the partition matrix B of the product-space clustering. By using the multi-view setting, many clusters benefit from the added information and are further refined. The enlarged area shows a uniform cluster that can now be divided into 2 distinct blocks.

In terms of the 3D structure (view 1), we could find as many as 9 clusters, whereas the amino acid sequence alignment (view 2) revealed 7 clusters, although presumably separating different types of groups. Since we want to highlight the benefit of multi-view clustering, we also construct the product space by concatenating the dimensions of both data matrices and treating it as a single view. This approach yields 8 stable clusters. Multi-view clustering is expected to further distinguish properties compared to the baseline of the product-space results. Indeed, in our experiment we receive a total number of 15 clusters, some of which show a clear refinement of already existing clusters. Figure 6.14 depicts one such case where product-space clustering identifies one seemingly uniform group of proteins and multi-view clustering further divides this into 2. From a biological standpoint, this is a feasible choice: One cluster only contains proteins that are responsible for negative

regulation of biological processes and the other contains proteins corresponding to positive regulation, as illustrated in Figure 6.15 and Figure 6.16. This demonstrates that dependency-seeking clustering is able to detect correlation in the data that is caused by a 'sign flip' of the biological processes. Figure 6.15 and Figure 6.16 were produced with "GOrilla", which is a "Gene Ontology enRIchment anaLysis and visuaLizAtion" tool. GOrilla identifies and visualizes enriched GO terms in ranked lists of genes.[2]

[2]http://cbl-gorilla.cs.technion.ac.il/.

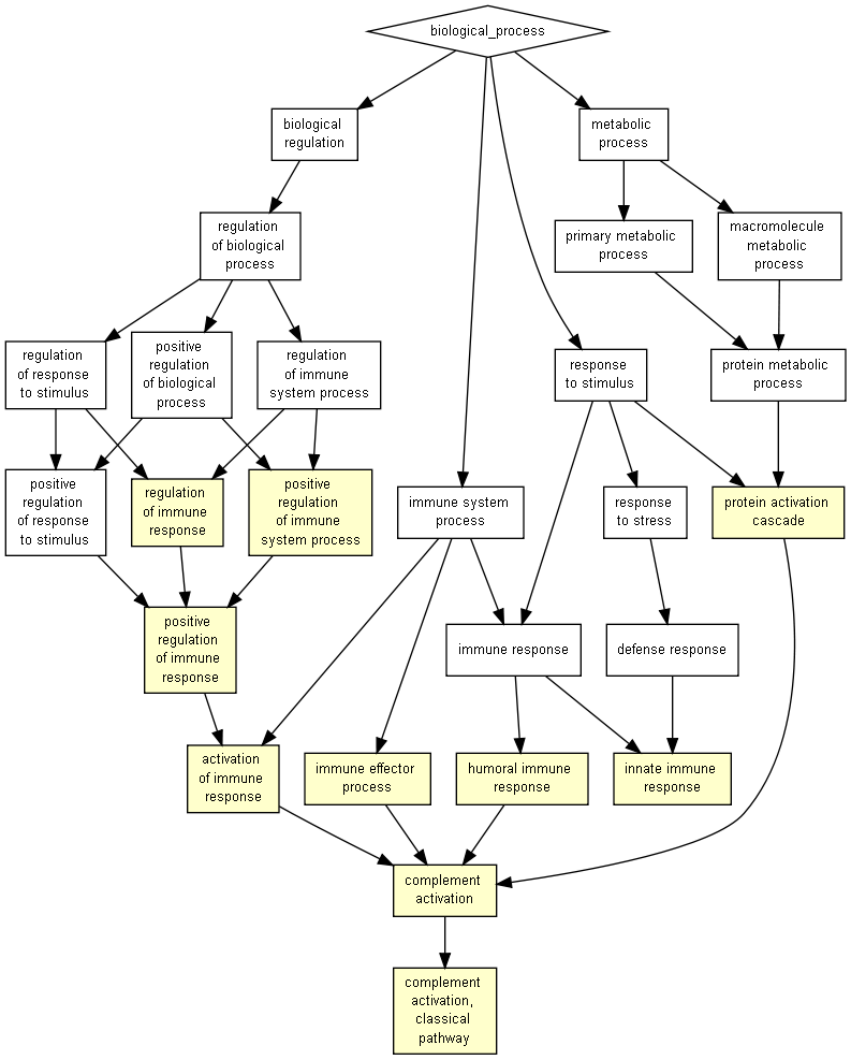

Figure 6.15: Biological process of proteins in cluster 8 found by multi-view clustering: The proteins define positive biological processes.

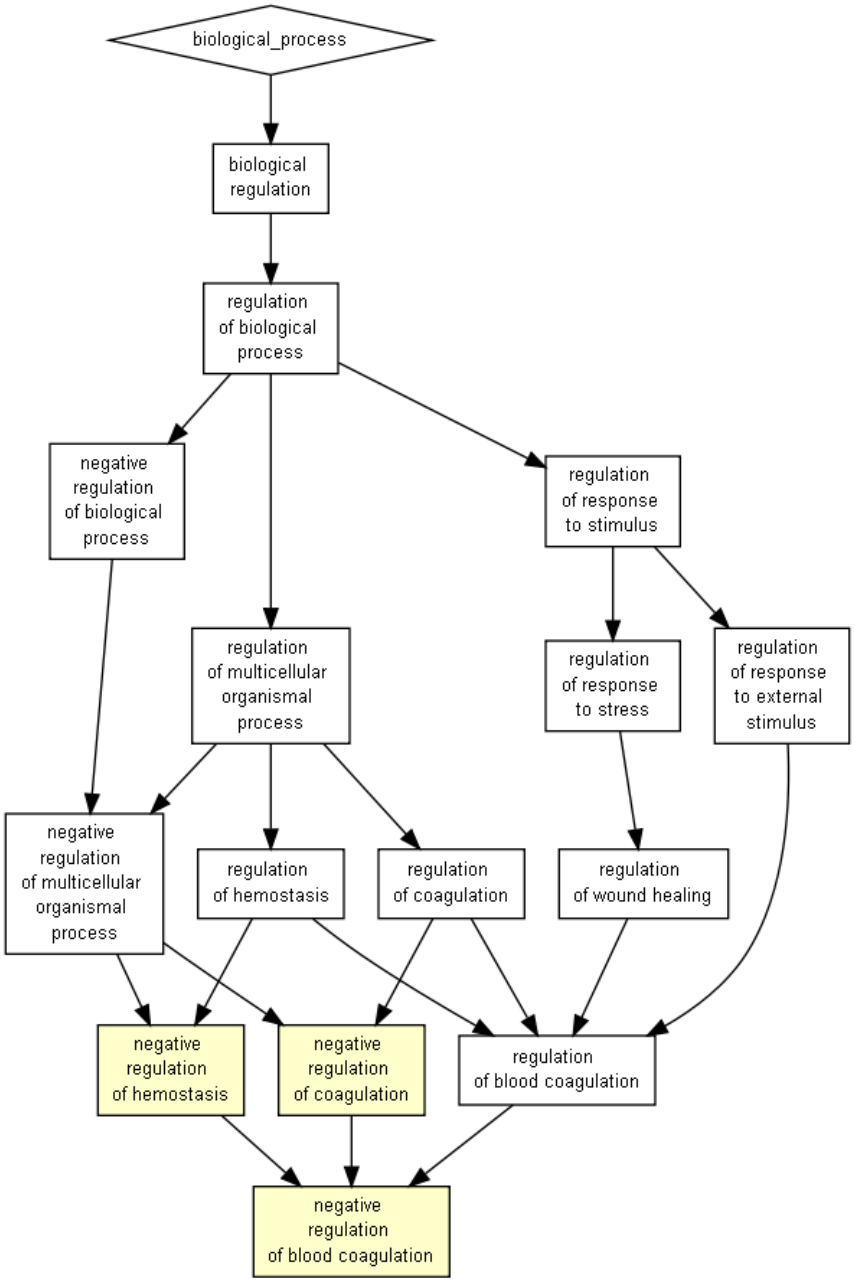

Figure 6.16: Biological process of proteins in cluster 9 found by multi-view clustering: The proteins define negative biological processes.

6.2.5 Outlook

The multi-view clustering model introduced above suffers from one serious limitation: the likelihood is only informative for $d < n$. For $d = n$ the covariance matrix Σ completely vanishes from the likelihood. In practice, the assumption $d < n$ is often not feasible. A possibility to avoid this problem consists in replacing the maximum likelihood estimate for Γ with a Bayesian estimate, as explained in the following.

We exploit that the matrix t-distribution is the distribution that results from the following marginal distribution (see e.g. [Kotz 04] for more details)

$$\int \mathcal{N}_{n,d}(X|\mathbf{0}, \Sigma, \Gamma)\mathcal{IW}_d(\Gamma|I_d)d\Gamma = t_{n,d}(\delta, \mathbf{0}, \Sigma, I_d) \tag{6.43}$$

where $t_{n,d}$ is a matrix-variate t-distribution with δ degrees of freedom and $\mathcal{IW}$ denotes the inverse Wishart distribution.

The matrix-variate t-distribution of a $n \times d$ matrix X is given by:

$$X|\Sigma \sim t_{n,d}(\mathbf{0}, \delta, \Sigma, I_d) \propto |\Sigma|^{-\frac{d}{2}}|I_n + \Sigma^{-1}XX^T|^{-\frac{\delta+n+d-1}{2}} \tag{6.44}$$

In the model introduced in 6.2.1, we used the maximum likelihood estimator for Γ and arrived at the likelihood 6.16:

$$l(\Sigma; X) = -\tfrac{d}{2}\log|\Sigma| - \tfrac{n}{2}\log|X^TWX|$$

The problematic part in this likelihood is the second term $\frac{n}{2}\log|X^TWX|$, as for $d = n$ it holds that $\frac{n}{2}\log|X^TWX| = \frac{n}{2}\log|W||XX^T|$.

By replacing the maximum likelihood estimator with the Bayesian estimator 6.44, we obtain the following likelihood:

$$l(\Sigma; X) = -\tfrac{d}{2}\log|\Sigma| - \tfrac{\delta+n+d-1}{2}\log|I_n + WXX^T| \tag{6.45}$$

The likelihood term $-\frac{\delta+n+d-1}{2}\log|I_n + WXX^T|$ will never split up, due to the added term I_n. Hence, by using the Bayesian estimator, the model is feasible for all X matrices independent of the rank of X.

Further work includes a thorough elaboration of the problem sketched above. The downside with this model is that sampling is very costly, as in every update the determinant of a $n \times n$ matrix has to be computed instead of a $d \times d$ matrix.

6.3 Summary

The first contribution in this chapter consists in introducing a very flexible probabilistic model for clustering dissimilarity data. It contains an exchangeable partition process prior which avoids label-switching problems. The likelihood component follows a generalized Wishart model for squared Euclidean distance matrices which is invariant under translations and rotations of the underlying coordinate system, under permutations of the object- and cluster- indices, and under scaling transformations. We call the final clustering model the *Translation Invariant Wishart-Dirichlet* (TIWD) cluster process. The main contributions in Section 6.1 are threefold:

(i) On the modeling side, we propose that it is better to work directly on the distances, without computing an explicit dot-product- or vector-space-representation, since such embeddings add unnecessary noise to the inference process. Experiments on simulated data corroborate this proposition by showing that the TIWD model significantly outperforms alternative approaches. In particular if the clusters are only poorly separated, the full probabilistic nature of the TIWD model has clear advantages over hierarchical approaches.

(ii) On the algorithmic side we show that costly matrix operations can be avoided by carefully exploiting the inner structure of the likelihood term. We prove that a sweep of a Gibbs sampler can be computed in $O(n^2 + nk_B^2)$ time, as opposed to $O(n^4 k_B)$ for a naive implementation. Experiments show that these algorithmic improvements make it possible to apply the model to large-scale data sets.

(iii) A semi-supervised experiment with globin proteins revealed the strength of our partition process model which is flexible enough to introduce new classes for objects which are dissimilar to any labeled observation. We could identify an interesting class of bacterial sequences, and a subsequent analysis of their domain structure showed that these sequences indeed share some unusual structural elements.

The second contribution in this chapter consists in the extension of the model to a transfer-learning scenario. Often, pairwise distances are not only observed in one, but in multiple views, resulting from different measurement techniques and/or different similarity measures. Hereby the term *view* refers to one realization of a distance matrix that gives a semantic meaning to the dimensions involved. The multi-view scenario can naturally be derived from the general likelihood when restricting the model to only allow a T-block diagonal correlation matrix Γ (Model B). Hereby, the likelihood splits into

$t = 1, \ldots, T$ separate terms, each responsible for an exclusive set of dimensions. The underlying assumption that the views are independent given the cluster structure forces the model to uncover dependencies between views. This process is called *Multi-View Translation Invariant Dirichlet* (MVTID) clustering process. The main contributions in Section 6.2 are the following:

i) Compared to the original TIWD model, the MVTID has dramatically increased degrees of freedom due to the new translation invariant likelihood and a full block matrix Σ. These changes enable the model to be flexible enough to cluster over multiple views and to detect dependencies between views. Synthetic experiments showed that the MVTID process implicitly expresses dependencies by introducing new clusters, and thus reveals hidden information. The straight-forward approach of clustering the product space of all views completely fails to achieve this, simply because it cannot detect dependencies between views. Even in cases where inter-view dependency is known to be non-existent, it is worse to cluster in the product space since at some point we definitely will violate the model assumption $d < n$ by simply concatenating many views. In contrast to this, multi-view clustering enables us to jointly work on theoretically arbitrary high numbers of views without ever exhausting the allowed range of d. In practice, one might also observe small signal to noise ratios in single views that quickly grow to be problematic for accumulation: Summing up noisy S matrices is prone to produce a matrix where the joint block structure is not visible anymore.

ii) In terms of complexity, our algorithm requires a cost of $\mathcal{O}(n^3)$ for preprocessing and $\mathcal{O}(n)$ for one full sweep of the Gibbs sampler, if we use a truncated Dirichlet process. This improved run time is achieved by utilizing the block structure of all matrices involved on the one hand and by exploiting the computational benefits of the likelihood in X.

iii) In a real world experiment on clustering proteases that are available in two views our multi-view clustering yields a clear refinement of already existing clusters. The refinement of the clusters makes sense from a biological point of view and this example illustrates that the multi-view clustering approach is able to detect hidden correlation between views.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

The lack of sufficient training data is the limiting factor for many machine learning techniques. If data is available for several different but related problems, transfer learning learning can be used to learn over many related data sets. In this thesis we introduced new approaches in the area of transfer learning, both for supervised and for unsupervised data analysis as well as for vectorial data and for pairwise distance data. In both areas, we introduced novel methods and efficient algorithms which are applicable for a broad range of applications. In summary, we made the following contributions:

- In the first part which deals with supervised learning problems we consider vectorial data. We filled an existing gap in the Group-Lasso research by introducing a complete analysis of the $\ell_{1,p}$ Group-Lasso for all p-norms. The proposed active set algorithm is applicable to all p-norms with $1 \leq p \leq \infty$. We presented a theoretical and empirical comparison of various Group-Lasso methods that yield solutions that are sparse on the group-level.

- The main theoretical contribution in Chapter 4 consist in a unified characterization of all $\ell_{1,p}$ Group-Lasso methods by way of subgradient calculus. A simple testing procedure is presented to check for completeness and uniqueness of solutions.

- For the unified active set algorithm, a convergence guarantee to the global optimum is given. The main technical contribution in this part consisted in the convergence proof of the proposed interval bisection.

- With these technical derivations, a complete comparison of all Group-Lasso methods in large-scale experiments was possible for the first time. Both, the prediction performance in a multi-task learning scenario and the interpretability of solutions was investigated on synthetic and real world data sets.

* In the second part which deals with unsupervised learning problems we consider the common problem of solely obtaining pairwise distances without access to an underlying vector space. We face the problem of clustering distance data and to perform transfer learning on distance data. The application areas cover any data sets in form of pairwise distances.

* First, we introduce a Bayesian clustering model that is able to cluster on distance data directly. By avoiding unnecessary and possibly noisy embeddings, better performance of the clustering is observed. By encoding the translation-invariance directly into the likelihood, the model is very flexible.

* The model is fully probabilistic in nature, i.e. as output one obtains samples from a probability distribution over partitions and not just one single clustering solution. We use a Dirichlet process prior to partition the data.

* On the algorithmic side, a highly efficient Gibbs sampling procedure that exploits the block structure of the partition process is presented.

* By introducing more flexibility into the covariances and by adapting the likelihood, a transfer learning approach is presented. The method is able to cluster multiple, co-occurring views of the same phenomenon and to reveal structures that are shared between these data sets.

* Finally, both clustering methods are tested on synthetic and real world data sets and the advantage of encoding the translation-invariance directly into the likelihood becomes obvious. Several hierarchical clustering methods are clearly outperformed by our new clustering method and in the multi-view scenario, dependencies between different views are revealed.

7.2 Future Work

Transfer learning constitutes an important research area in the field of machine learning. In this thesis, some of the problems occurring in transfer learning are approached, but still many open questions remain. Further directions of research in multi-task learning include for instance the use of the so-called "0-norm" for inducing sparsity in Group-Lasso methods. The 0-norm is defined as the sum over the non-zero entries of x, i.e. $||x||_0 = \#\{x_i : x_i \neq 0\}$. Mathematically, however, the 0-norm is not a norm and the resulting problem is not convex. It was shown in [Moha 12] that by using a spike-and-slab prior that matches the 0-norm, better prediction performance was obtained than by using the ℓ_1 norm. In a Group-Lasso setting it is still an open question if it is feasible to use a $\ell_{q,p}$ Group-Lasso for multi-task learning for $0 \leq q < 1$ and $1 \leq p \leq \infty$. Many problems arise. For instance, the problem is not convex anymore, hence local minima can exist. Moreover, for $p \to 0$ the problem becomes a discrete optimization problem.

The second important aspect we approached in this thesis is the problem of learning on distance data directly without explicit embeddings into vector spaces. Many extensions of our TIWD model are imaginable. The development of new machine learning methods based on distance data are of high importance for many areas of application, especially in the biomedical field. Concrete extensions of methods of this kind are planned. Examples include a clustering method that allows to model overlapping clusters: so far, the basic approach is to cluster data into mutually exclusive partitions. In many applications, however, it is more realistic that data points may belong to multiple, overlapping clusters. If, for instance, a gene has many different functions, it might belong to more than one cluster. The aim is to build a model for overlapping clusters where the objects are available as distance data. The crucial part here is that instead of a Dirichlet process prior (a.k.a. Chinese Restaurant Process prior) on a partition matrix defining a partition process, a Beta-Binomial prior (a.k.a. Indian Buffet Process) is used. By abandoning the block structure, variational approximation methods ([Bish 09]) need to be developed to obtain an efficient algorithm that is suitable for high-dimensional data sets.

A further interesting extension consists in inferring networks directly from distance data. The idea is to use the translation-invariant Wishart likelihood, however, instead of partitioning the data by using a Dirichlet process prior, a Bayesian selection prior is used to infer sparsely connected networks. A suitable prior construction has to be chosen among the rich class of distributions over symmetric positive definite matrices that allows for network inference.

The prior should be flexible enough to provide a unique parametrization of the correlation matrix and to allow unconstrained values on the interval $(-1, 1)$. However, sampling models of this kind suffer from high computational costs and are hardly applicable to large networks. Therefore the usefulness of variational approximations and expectation propagation methods [Mink 01] have to be investigated. Additionally, as an alternative approach to network inference models within a Bayesian framework, classical neighborhood selection techniques based on lasso estimators might be investigated and extended to distance data. The idea is to use a penalized Wishart likelihood for network inference on distance data.

The extensions mentioned so far rely on static data. However, often data is obtained at different points in time and dynamic models that take a time component into account are needed. Frequently in biomedical applications, genes are measured at different points in time, for instance in order to examine the efficiency of a medication over time. In such situations it is important to generalize network inference methods to account for possible time variations in the association structure. Time-varying network inference of this kind for vectorial data has been proposed for instance in [Kola 12] and in [Zhou 10]. Hence, as a further extension, the problem of dynamic network inference on distance data might be investigated. A model has to be developed that is not only able to recover networks from distance data but also from distance data that arrives in different epochs. The aim is to estimate time-varying networks from distance data.

As mentioned in Section 5.3, pairwise distances are obtained for example from string alignment scores or from Mercer kernels. Mercer kernels can encode similarities between many different kinds of objects as for instance kernels on graphs, images, distributions, structures or strings. Hence, the clustering methods proposed in Chapter 6 as well as the possible extensions mentioned here cover a broad scope of application, not only in the biomedical field but in a variety of fields where distance- or kernel matrices are obtained.

Bibliography

[Ahme 08] A. Ahmed and E. Xing. "Dynamic Non-Parametric Mixture Models and The Recurrent Chinese Restaurant Process: with Applications to Evolutionary Clustering". *Proceedings of The Eighth SIAM International Conference on Data Mining (SDM)*, 2008.

[Ande 46] T. Anderson. "The Non-Central Wishart Distribution and Certain Problems of Multivariate Statistics". *Ann. Math. Statist.*, Vol. 17, No. 4, pp. 409–431, 1946.

[Argy 07] A. Argyriou, T. Evgeniou, and M. Pontil. "Multi-task feature learning". In: *Advances in Neural Information Processing Systems 19*, MIT Press, 2007.

[Atti 00] H. Attias. "A variational Bayesian framework for graphical models". *Advances in Neural Information Processing Systems 12*, pp. 209–215, 2000.

[Bach 08] F. Bach. "Consistency of the group Lasso and multiple kernel learning". *JMLR*, Vol. 9, pp. 1179–1225, 2008.

[Bela 07] M. Belabbas and P. Wolfe. "Fast low-rank approximation for covariance matrices". In: *IEEE Workshop on Computational Advances in Multi-Sensor Processing*, pp. 293 – 296, 2007.

[Bert 95] D. P. Bertsekas. *Nonlinear programming.* Athena Scientific, 1995.

[Bick 04] S. Bickel and T. Scheffer. "Multi-View Clustering". In: *Proceedings of the IEEE International Conference on Data Mining*, 2004.

[Bick 05] S. Bickel and T. Scheffer. "Estimation of mixture models using Co-EM". In: *In Proceedings of the ICML Workshop on Learning with Multiple Views*, 2005.

[Bick 08] S. Bickel, J. Bogojeska, T. Lengauer, and T. Scheffer. "Multi-task learning for HIV therapy screening". In: *Proceedings of the 25th international conference on Machine learning*, pp. 56–63, 2008.

[Bilo 99] M. Bilodeau and D. Brenner. *Theory of Multivariate Statistics*. Springer, 1999.

[Bish 09] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2009.

[Blei 06] D. Blei and M. Jordan. "Variational inference for Dirichlet process mixtures". *Bayesian Analysis*, Vol. 1, pp. 121–144, 2006.

[Blei 11] D. M. Blei and P. Frazier. "Distance dependent Chinese restaurant processes". *Journal of Machine Learning Reseach*, No. 12, pp. 2461–2488, 2011.

[Brei 01] L. Breiman. "Random Forests". *Machine Learning*, Vol. 45, pp. 5–32, 2001.

[Brow 86] L. D. Brown. *Fundamentals of statistical exponential families: with applications in statistical decision theory*. Institute of Mathematical Statistics, Hayworth, CA, USA, 1986.

[Caru 97] R. Caruana. "Multitask Learning". In: *Machine Learning*, pp. 41–75, 1997.

[Cent 06] T. P. Centeno and N. Lawrence. "Optimising kernel parameters and regularisation coefficients for non-linear discriminant analysis". *Journal of Machine Learning Research*, Vol. 7, No. 455-49, 2006.

[Chau 09] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan. "Multi-View Clustering via Canonical Correlation Analysis". In: *ICML*, 2009.

[Cili 05] R. Cilibrasi and P. Vitanyi. "Clustering by compression". *IEEE Transactions on Information Theory*, Vol. 51, No. 4, pp. 1523–1545, April 2005.

[Cord 01] A. Corduneanu and C. M. Bishop. "Variational bayesian model selection for mixture distributions". *Eighth International Workshop on Artifficial Intelligence and Statistics*, pp. 27–34, 2001.

[Cox 01] T. Cox and M. Cox. *Multidimensional Scaling*. Chapman & Hall, 2001.

[Croo 04] G. Crooks, G. Hon, J. Chandonia, and S. Brenner. "WebLogo: A sequence logo generator". *Genome Research*, Vol. 14, No. 1188-1190, 2004.

[Dahi 07] C. Dahinden, G. Parmigiani, M. Emerick, and P. Bühlmann. "Penalized likelihood for sparse contingency tables with an application to full-length cDNA libraries". *BMC Bioinformatics*, Vol. 8, p. 476, 2007.

[Dahl 05] D. Dahl. "Sequentially-allocated merge-split sampler for conjugate and non-conjugate Dirichlet process mixture models". Tech. Rep., Department of Statistics, Texas A&M University, 2005.

[Dani 09] M. J. Daniels and M. Pourahmadi. "Modeling covariance matrices via partial autocorrelations". *J. Multivariate Analysis*, Vol. 100, No. 10, pp. 2352–2363, 2009.

[Demp 77] A. Dempster, N. Laird, and D. Rubin. "Maximum likelihood from incomplete data via the EM algorithm". *Journal of the Royal Statistical Society series B*, Vol. 39, pp. 1–38, 1977.

[Dill 11] M. T. Dill, F. H. Duong, J. E. Vogt, S. Bibert, P.-Y. Bochud, L. Terracciano, A. Papassotiropoulos, V. Roth, and M. H. Heim. "Interferon-Induced Gene Expression is a Stronger Predictor of Treatment Response Than IL28B Genotype in Patients With Hepatitis C". *Gastroenterology*, pp. 1021–1031.e10, 2011.

[Dubr 01] A. Dubrulle. "Retooling the method of block conjugate gradients". *Electronic Transactions on Numerical Analysis*, Vol. 12, pp. 216–233, 2001.

[Dunn 61] O. J. Dunn. "Multiple comparisons among means". *JASA*, Vol. 56, pp. 54–64, 1961.

[Efro 03] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. "Least Angle Regression". Tech. Rep., Statistics Department, Stanford University, 2003.

[Efro 04] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. "Least Angle Regression". *Ann. Stat.*, Vol. 32, pp. 407–499, 2004.

[Ewen 72] W. Ewens. "The sampling theory of selectively neutral alleles". *Theoretical Population Biology*, Vol. 3, pp. 87–112, 1972.

[Ferg 73] T. S. Ferguson. "A Bayesian Analysis of Some Nonparametric Problems". *The Annals of Statistics*, Vol. 1, No. 2, pp. pp. 209–230, 1973.

[Fisc 04] B. Fischer, V. Roth, and J. M. Buhmann. "Clustering with the Connectivity Kernel". In: S. Thrun, L. Saul, and B. Schölkopf, Eds., *Advances in Neural Information Processing Systems 16*, pp. 89–96, MIT Press, Cambridge, MA, 2004.

[Fors 10] W. Forst and D. Hoffmann. *Optimization - Theory and Practice*. Springer, 2010.

[Frey 07] B. J. Frey and D. Dueck. "Clustering by Passing Messages Between Data Points". *Science*, Vol. 315, pp. 972–976, 2007.

[Frie 07] J. Friedeman, T. Hastie, and R. Tibshirani. "Sparse inverse covariance estimation with the Graphical Lasso". *Biostatistics*, No. 9, pp. 432–441, 2007.

[Golu 89] G. Golub and C. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD, USA, 1989.

[Gorn 11] N. Görnitz, C. Widmer, G. Zeller, A. Kahles, S. Sonnenburg, and G. Rätsch. "Hierarchical Multitask Structured Output Learning for Large-scale Sequence Segmentation". In: *Advances in Neural Information Processing Systems*, 2011.

[Gran 98] Y. Grandvalet. "Least absolute shrinkage is equivalent to quadratic penalization". In: L. Niklasson, M. Bodén, and T. Ziemske, Eds., *ICANN'98*, pp. 201–206, Springer, 1998.

[Gupt 00] A. K. Gupta and D. K. Nagar. *Matrix Variate Distributions*. Chapman & Hall, 2000.

[Hast 94] T. Hastie, R. Tibshirani, and A. Buja. "Flexible discriminant analysis by optimal scoring". *J. American Statistical Association*, Vol. 89, pp. 1255–1270, 1994.

[Hast 96] T. Hastie and R. Tibshirani. "Discriminant analysis by Gaussian mixtures". *J. Royal Statistical Society series B*, Vol. 58, pp. 158–176, 1996.

[Hast 98] T. Hastie and R. Tibshirani. "Classification by pairwise coupling". In: M. I. Jordan, M. J. Kearns, and S. A. Solla, Eds., *Advances in Neural Information Processing Systems*, The MIT Press, 1998.

[Hofm 97] T. Hofmann and J. Buhmann. "Pairwise Data Clustering by Deterministic Annealing". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 1, pp. 1–14, 1997.

[Jain 88] A. Jain and R. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[Jala 10] A. Jalali, P. Ravikumar, S. Sanghavi, and C. Ruan. "A Dirty Model for Multi-task learning". NIPS, 2010.

[Jasr 05] A. Jasra, C. C. Holmes, and D. A. Stephens. "Markov Chain Monte Carlo Methods and the Label Switching Problem in Bayesian Mixture Models". *Statistical Science*, Vol. 20, No. 1, p. 50 :67, 2005.

[Joe 96] H. Joe. "Families of m-variate distributions with given margins and m(m-1)/2 bivariate dependence parameters". In: L. Rüschendorf, B. Schweizer, and M. Taylor, Eds., *Distributions with Fixed Marginals and Related Topics*, pp. 120–141, AMS, 1996.

[Kim 06] Y. Kim, J. Kim, and Y. Kim. "Blockwise Sparse Regression". *Statistica Sinica*, Vol. 16, pp. 375–390, 2006.

[Klam 06] A. Klami and S. Kaski. "Generative Models that Discover Dependencies Between Data Sets". In: *Machine Learning for Signal Processing, 2006. Proceedings of the 2006 16th IEEE Signal Processing Society Workshop on*, pp. 123 –128, sept. 2006.

[Klam 08] A. Klami. *Modeling of mutual dependencies.* PhD thesis, Helsinki University of Technology, 2008.

[Kola 12] M. Kolar, L. Song, A. Ahmed, and E. P. Xing. "Estimating time-varying networks". *Ann. Appl. Stat*, Vol. 4, No. 1, pp. 94–123, 2012.

[Kotz 04] S. Kotz and S. Nadarajah. *Multivariate t Distributions and Their Applications*. Cambridge University Press, 2004.

[Kuma 96] N. Kumar and A. Andreou. "Generalization of linear discriminant analysis in a maximum likelihood framework". In: *Proc. Joint Meeting of the American Statistical Association*, 1996.

[Lanc 04] G. Lanckriet, M. Deng, N. Cristianini, M. Jordan, and W. Noble. "Kernel-based data fusion and its application to protein function prediction in yeast". In: *Pacific Symposium on Biocomputing*, pp. 300–311, 2004.

[Lang 03] T. Lange, M. L. Braun, V. Roth, and J. M. Buhmann. "Stability-Based Model Selection". In: S. T. S. Becker and K. Obermayer, Eds., *Advances in Neural Information Processing Systems 15*, pp. 617–624, MIT Press, Cambridge, MA, 2003.

[Liu 09] H. Liu, M. Palatucci, and J. Zhang. "Blockwise Coordinate Descent Procedures for the Multi-task Lasso, with Applications to Neural Semantic Basis Discovery". 26th Intern. Conference on Machine Learning, 2009.

[Liu 10a] J. Liu and J. Ye. "Efficient ℓ_1/ℓ_q Norm Regularization". Tech. Rep., 2010.

[Liu 10b] Q. Liu, Q. Xu, V. W. Zheng, H. Xue, Z. Cao, and Q. Yang. "Multi-task learning for cross-platform siRNA efficacy prediction: an in-silico study". *BMC Bioinformatics*, Vol. 11, No. 1, p. 181, 2010.

[MacE 94] S. MacEachern. "Estimating normal means with a conjugate-style Dirichlet process prior". *Communication in Statistics: Simulation and Computation*, Vol. 23, pp. 727–741, 1994.

[MacK 95] D. MacKay. "Probable networks and plausible predictions - a review of practical Bayesian methods for supervised neural networks". *Network: Computation in Neural Systems*, Vol. 6, pp. 469–505, 1995.

[McCu 08a] P. McCullagh. "Marginal Likelihood for Parallel series". *Bernoulli*, Vol. 14, pp. 593–603, 2008.

[McCu 08b] P. McCullagh and J. Yang. "How many clusters?". *Bayesian Analysis*, Vol. 3, pp. 101–120, 2008.

[McCu 09] P. McCullagh. "Marginal Likelihood for Distance Matrices". *Statistica Sinica*, Vol. 19, pp. 631–649, 2009.

[McCu 83] P. McCullagh and J. Nelder. *Generalized Linear Models*. Chapman & Hall, 1983.

[Meie 06] L. Meier, S. van de Geer, and P. Bühlmann. "The Group Lasso for Logistic Regression". Tech. Rep. 131, ETH Zurich, 2006.

[Meie 08] L. Meier, S. van de Geer, and P. Bühlmann. "The Group Lasso for Logistic Regression". *J. Roy. Stat. Soc. B*, Vol. 70, No. 1, pp. 53–71, 2008.

[Mein 06] N. Meinshausen and P. Bühlmann. "High dimensional graphs and variable selection with the Lasso". *Annals of Statistics*, Vol. 34, pp. 1436–1462, 2006.

[Micc 05] C. A. Micchelli and M. Pontil. "Learning the kernel function via regularization". *Journal of Machine Learning Research*, Vol. 6, pp. 1099–1125, 2005.

[Mink 01] T. P. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

[Moha 12] S. Mohamed, K. A. Heller, and Z. Ghahramani. "Bayesian and L_1 Approaches for Sparse Unsupervised Learning". *ICML*, 2012.

[Neal 00] R. Neal. "Markov chain sampling methods for Dirichlet process mixture models". *Journal of Computational and Graphical Statistics*, Vol. 9, pp. 249–265, 2000.

[Ng 01] A. Y. Ng, M. I. Jordan, and Y. Weiss. "On Spectral Clustering: Analysis and an algorithm". In: *Advances in Neural Information Processing Systems 14*, pp. 849–856, MIT Press, 2001.

[Nils 07] R. Nilsson, J. Peña, J. Björkegren, and J. Tegnér. "Consistent Feature Selection for Pattern Recognition in Polynomial Time". *JMLR*, Vol. 8, pp. 589–612, 2007.

[Oboz 06] G. Obozinski and B. Taskar. "Multi-task feature selection". *In the workshop of structural Knowledge Transfer for Machine Learning in the 23rd International Conference on Machine Learning*, 2006.

[Osbo 00] M. Osborne, B. Presnell, and B. Turlach. "On the LASSO and its dual". *J. Comp. and Graphical Statistics*, Vol. 9, No. 2, pp. 319–337, 2000.

[Pitm 06] J. Pitman. "Combinatorial Stochastic Processes". In: J. Picard, Ed., *Ecole d'Ete de Probabilites de Saint-Flour XXXII-2002*, Springer, 2006.

[Pres 07] W. H. Press, S. A. Teukolsky, W. Vetterling, and B. Flannery. *NUMERICAL RECIPES. The Art of Scientific Computing.* Vol. third edition, Cambridge University Press, 2007.

[Quat 09] A. Quattoni, X. Carreras, M. Collins, and T. Darrell. "An Efficient Projection for $l_{1\infty}$ Regularization". 26th Intern. Conference on Machine Learning, 2009.

[Rats 04] G. Rätsch and S. Sonnenburg. "Accurate splice site detection for Caenorhabditis elegans". *Kernel Methods in Computational Biology*, pp. 277–298, 2004.

[Rats 05] G. Rätsch, S. Sonnenburg, and B. Schölkopf. "RASE: recognition of alternatively spliced exons in C. elegans". *Bioinformatics*, Vol. 21, pp. i369–i377, 2005.

[Robe 05] C. Robert and G. Casella. *Monte Carlo Statistical Methods.* Springer, 2005.

[Rose 08] M. Rosen-Zvi, A. Altmann, M. Prosperi, E. Aharoni, H. Neuvirth, A. Sonnerborg, E. Schulter, D. Struck, Y. Peres, F. Incardona, R. Kaiser, M. Zazzi, and T. Lengauer. "Selecting anti-HIV therapies based on a variety of genomic and clinical factors". *Bioinformatics*, Vol. 24, pp. i399–i406, 2008.

[Roth 03a] V. Roth, J. Laub, J. Buhmann, and K.-R. Müller. "Going Metric: Denoising Pairwise Data". In: S. T. S. Becker and K. Obermayer, Eds., *Advances in Neural Information Processing Systems 15*, pp. 817–824, MIT Press, Cambridge, MA, 2003.

[Roth 03b] V. Roth, J. Laub, M. Kawanabe, and J. Buhmann. "Optimal Cluster Preserving Embedding of Non-Metric Proximity Data". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 12, 2003.

[Roth 04] V. Roth. "The Generalized LASSO". *IEEE Trans. Neural Networks*, Vol. 15, No. 1, pp. 16–28, 2004.

[Roth 07] V. Roth and B. Fischer. "Improved Functional Prediction of Proteins by Learning Kernel Combinations in Multilabel Settings". *BMC Bioinformatics*, Vol. 8, No. Suppl.2, 2007.

[Roth 08] V. Roth and B. Fischer. "The Group-Lasso for generalized linear models: uniqueness of solutions and efficient algorithms". In: *ICML '08*, pp. 848–855, 2008.

[Schm 08] M. Schmidt and K. Murphy. "Structure learning in random fields for heart motion abnormality detection". In: *In CVPR*, 2008.

[Scho 97] B. Schölkopf, A. Smola, and K.-R. Müller. "Kernel Principal Component Analysis". *Artificial Neural Networks: ICANN*, 1997.

[Shev 03] K. Shevade and S. Keerthi. "A simple and efficient algorithm for gene selection using sparse logistic regression". *Bioinformatics*, Vol. 19, pp. 2246–2253, 2003.

[Sing 02] D. Singh, P. Febbo, K. Ross, D. Jackson, J. Manola, C. Ladd, P. Tamayo, A. Renshaw, D'Amico, J. Richie, E. Lander, M. Loda, P. Kantoff, T. Golub, and W. Sellers. "Gene expression correlates of clinical prostate cancer behavior". *Cancer Cell*, Vol. 1, No. 2, pp. 203–209, March 2002.

[Spel 98] P. Spellman, G. Sherlock, M. Zhang, V. Iyer, K. Anders, M. Eisen, P. Brown, D. Botstein, and B. Futcher. "Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization.". *Mol Biol Cell.*, Vol. 9, No. 12, pp. 3273–97, Dec 1998.

[Sriv 03] M. Srivastava. "Singular Wishart and multivariate beta distributions". *Annals of Statistics*, Vol. 31, No. 2, pp. 1537–1560, 2003.

[Stre 09] A. P. Streich, M. Frank, and J. M. Buhmann. "Multi-Assignment Clustering for Boolean Data". *ICML*, 2009.

[Tann 96] A. Tannapfel, H. A. Hahn, A. Katalinic, R. J. Fietkau, R. Kuhn, and C. W. Wittekind. "Prognostic value of ploidy and proliferation markers in renal cell carcinoma.". *Cancer*, Vol. Jan 1;77(1), pp. 164–71, 1996.

[Tibs 96] R. Tibshirani. "Regression shrinkage and selection via the Lasso". *J. Roy. Stat. Soc. B*, Vol. 58, No. 1, pp. 267–288, 1996.

[Torg 58] W. Torgerson. *Theory and Methods of Scaling*. John Wiley and Sons, New York, 1958.

[Turl 05] B. A. Turlach, W. N. Venables, and S. J. Wright. "Simultaneous Variable Selection". *Technometrics*, Vol. 47, No. 349-363, 2005.

[UniP 10] UniProt Consortium. "The Universal Protein Resource (UniProt) in 2010". *Nucleic Acids Res.*, Vol. D142-D148, 2010.

[Vemp 04] S. Vempala. *The Random Projection Method. Series in Discrete Mathematics and Theoretical Computer Science*, AMS, 2004.

[Vogt 10a] J. E. Vogt, S. Prabhakaran, T. J. Fuchs, and V. Roth. "The Translation-invariant Wishart-Dirichlet Process for Clustering Distance Data". In: *ICML*, pp. 1111–1118, 2010.

[Vogt 10b] J. E. Vogt and V. Roth. "The Group-Lasso: $\ell_{1,\infty}$ Regularization versus $\ell_{1,2}$ Regularization". In: *DAGM 2010*, pp. 252–261, Springer, 2010.

[Vogt 12] J. E. Vogt and V. Roth. "A Complete Analysis of the $\ell_{1,p}$ Group-Lasso". *ICML*, 2012.

[Wain 05] M. Wainwright, T. Jaakkola, and A. Willsky. "A New Class of Upper Bounds on the Log Partition Function". *IEEE Trans. Information Theory*, Vol. 51, No. 7, 2005.

[Wedd 73] R. W. M. Wedderburn. "On the Existence and Uniqueness of the Maximum Likelihood Estimates for Certain Generalized Linear Models". *Biometrika*, Vol. 63, No. 1, pp. 27–32, 1973.

[Wels 01] J. B. Welsh, L. M. Sapinoso, A. I. Su, S. G. Kernand, J. Wang-Rodriguez, C. A. Moskaluk, H. F. Frierson, and G. M. Hampton. "Analysis of Gene Expression Identifies Candidate Markers and Pharmacological Targets in Prostate Cancer". *Cancer Research*, Vol. 61, No. 16, p. 5974 :5978, August 2001.

[Widm 10] C. Widmer, N. Toussaint, Y. Altun, and G. Rätsch. "Inferring Latent Task Structure for Multi-Task Learning by Multiple Kernel Learning". *BMC Bioinformatics*, Vol. 11, No. Suppl. 8, p. S5, 2010.

[Widm 12] C. Widmer and G. Rätsch. "Multitask Learning in Computational Biology". In: *ICML 2011 Unsupervised and Transfer Learning Workshop*, 2012.

[Xue 07] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. "Multi-task learning for classification with dirichlet process priors". *Journal of Machine Learning Research*, Vol. 8, p. 2007, 2007.

[Yeo 04] G. Yeo and C. Burge. "Maximum entropy modeling of short sequence motifs with applications to RNA splicing signals". *J. Comp. Biology*, Vol. 11, pp. 377–394, 2004.

[Yosh 02] H. Yoshimoto, K. Saltsman, A. Gasch, H. Li, N. Ogawa, D. Botstein, P. Brown, and M. Cyert. "Genome-wide analysis of gene expression regulated by the calcineurin/Crz1p signaling pathway in Saccharomyces cerevisiae". *J Biol Chem.*, Vol. 277, No. 34, pp. 31079–88, Aug 2002.

[Yu 06] S. Yu. *Advanced Probabilistic Models for Clustering and Projection.* PhD thesis, University of Munich, 2006.

[Yu 07] S. Yu, V. Tresp, and K. Yu. "Robust multi-task learning with t-processes". In: *Proceedings of the 24th international conference on Machine learning*, pp. 1103–1110, ACM, New York, NY, USA, 2007.

[Yuan 06] M. Yuan and Y. Lin. "Model Selection and Estimation in Regression with Grouped Variables". *J. Roy. Stat. Soc. B*, pp. 49–67, 2006.

[Yver 03] G. Yvert, R. Brem, J. Whittle, J. Akey, E. Foss, E. Smith, R. Mackelprang, and L. Kruglyak. "Trans-acting regulatory variation in Saccharomyces cerevisiae and the role of transcription factors". *Nature Genet.*, Vol. 35, No. 1, pp. 57–64, Sep 2003.

[Zhan 10] Y. Zhang, D. Yeung, and Q. Xu. "Probabilistic Multi-Task Feature Selection". NIPS, 2010.

[Zhou 10] S. Zhou, J. Lafferty, and L. Wasserman. "Time Varying Undirected Graphs". *Machine Learning*, Vol. 80, No. 2-3, pp. 295–319, 2010.

[Zhu 05] X. Zhu, Z. Ghahramani, and J. Lafferty. "Time-Sensitive Dirichlet Process Mixture Models". Tech. Rep., 2005.

[Zou 05] H. Zou and T. Hastie. "Regularization and variable selection via the elastic net". *J.R. Statist. Soc. B*, Vol. 67, pp. 301–320, 2005.