TKK Dissertations in Information and Computer Science
Espoo 2008 TKK-ICS-D6

MODELING OF MUTUAL DEPENDENCIES

Arto Klami

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Faculty of Information and Natural Sciences for public examination and debate in Auditorium T2 at Helsinki University of Technology (Espoo, Finland) on the 5th of September, 2008, at 12 o'clock noon.

Helsinki University of Technology
Faculty of Information and Natural Sciences
Department of Information and Computer Science

Teknillinen korkeakoulu
Informaatio- ja luonnontieteiden tiedekunta
Tietojenkäsittelytieteen laitos

Distribution:
Helsinki University of Technology
Faculty of Information and Natural Sciences
Department of Information and Computer Science
P.O.Box 5400
FI-02015 TKK
FINLAND
Tel. +358-9-451 3272
Fax. +358-9-451 3277
Email: series@ics.tkk.fi

ISBN 978-951-22-9519-7 (Print)
ISBN 978-951-22-9520-3 (Online)
ISSN 1797-5050 (Print)
ISSN 1797-5069 (Online)
URL: http://lib.tkk.fi/Diss/2008/isbn9789512295203/

Multiprint Oy
Espoo 2008

ABSTRACT

Klami, A. (2008): **Modeling of mutual dependencies.** Doctoral thesis, Helsinki University of Technology, Dissertations in Information and Computer Science, TKK-ICS-D6, Espoo, Finland.

Keywords: canonical correlation analysis, clustering, data fusion, exploratory data analysis, probabilistic modeling, learning metrics, mutual dependency, mutual information

Data analysis means applying computational models to analyzing large collections of data, such as video signals, text collections, or measurements of gene activities in human cells. Unsupervised or exploratory data analysis refers to a subtask of data analysis, in which the goal is to find novel knowledge based on only the data. A central challenge in unsupervised data analysis is separating relevant and irrelevant information from each other. In this thesis, novel solutions to focusing on more relevant findings are presented.

Measurement noise is one source of irrelevant information. If we have several measurements of the same objects, the noise can be suppressed by averaging over the measurements. Simple averaging is, however, only possible when the measurements share a common representation. In this thesis, we show how irrelevant information can be suppressed or ignored also in cases where the measurements come from different kinds of sensors or sources, such as video and audio recordings of the same scene.

For combining the measurements, we use mutual dependencies between them. Measures of dependency, such as mutual information, characterize commonalities between two sets of measurements. Two measurements can hence be combined to reduce irrelevant variation by finding new representations for the objects so that the representations are maximally dependent. The combination is optimal, given the assumption that what is in common between the measurements is more relevant than information specific to any one of the sources.

Several practical models for the task are introduced. In particular, novel Bayesian generative models, including a Bayesian version of the classical method of canonical correlation analysis, are given. Bayesian modeling is especially justified approach to learning from small data sets. Hence, generative models can be used to extract dependencies in a more reliable manner in, for example, medical applications, where obtaining a large number of samples is difficult. Also, novel non-Bayesian models are presented: Dependent component analysis finds linear projections which capture more general dependencies than earlier methods.

Mutual dependencies can also be used for supervising traditional unsupervised learning methods. The learning metrics principle describes how a new distance metric focusing on relevant information can be derived based on the dependency between the measurements and a supervising signal. In this thesis, the approximations and optimization methods required for using the learning metrics principle are improved.

TIIVISTELMÄ

Klami, A. (2008): **Tietoaineistojen välisten riippuvuuksien mallintaminen.** Väitöskirja, Teknillinen korkeakoulu, Dissertations in Information and Computer Science, TKK-ICS-D6, Espoo, Suomi.

Avainsanat: aineistojen välinen riippuvuus, kanoninen korrelaatioanalyysi, klusterointi, oppiva metriikka, probabilistinen mallintaminen, tietolähteiden yhdistäminen, tutkiva data-analyysi, yhteisinformaatio

Data-analyysissä tutkitaan laskennallisia menetelmiä joilla voidaan tulkita suuria tietoaineistoja, kuten videosignaaleja, tekstikokoelmia tai ihmisen geenien aktiivisuusmittauksia. Ohjaamattomalla tai tutkivalla analyysilla tarkoitetaan data-analyysin alalajia, jossa tavoitteena on löytää uutta tietoa pelkästään annetun tietoaineiston perusteella. Eräs ohjaamattoman data-analyysin keskeisistä haasteista on hyödyllisen ja hyödyttömän informaation erottaminen toisistaan. Tässä väitöskirjassa esitetään uusia ratkaisuja hyödyllisiin löydöksiin keskittymiseksi.

Mittausvirheet ovat eräs hyödyttömän informaation lähde. Jos käytössä on useita mittauksia samasta kohteesta, voidaan mittausvirheiden vaikutusta vähentää ottamalla mittauksista keskiarvo. Tämä lähestymistapa on kuitenkin mahdollinen vain silloin, kun mittaukset on esitetty keskenään samalla tavalla. Tässä väitöskirjassa osoitetaan kuinka hyödytöntä informaatiota voidaan karsia myös tapauksissa, joissa mittaukset on saatu erilaisista lähteistä. Väitöskirjassa esitetyllä tavalla voidaan yhdistää erimerkiksi samasta kohteesta tallennettua video- ja äänisignaalia.

Mittausten yhdistämiseen käytetään aineistojen välisiä riippuvuuksia. Riippuvuudella, jota voidaan mitata esimerkiksi yhteisinformaatiolla, voidaan luonnehtia kahden aineiston välisiä yhtäläisyyksiä. Kaksi aineistoa voidaankin siis yhdistää hyödyllisemmän informaation korostamiseksi etsimällä niille uudet mahdollisimman paljon toisistaan riippuvat esitykset. Jos oletamme, että aineistojen väliset yhtäläisyydet ovat kiinnostavampia kuin yhdelle aineistoille ominaiset piirteet, on tällainen yhdistämistapa paras mahdollinen.

Väitöskirjassa esitellään useita menetelmiä joilla kyseinen tehtävä voidaan ratkaista käytännössä. Erityisesti työssä esitellään uusia Bayesilaisia generatiivisia malleja riippuvuuksien etsimiseksi. Eräs näistä on Bayesilainen versio kanonisesta korrelaatioanalyysista. Bayesilainen mallintaminen on erityisen perusteltua pieniä tietoaineistoja analysoitaessa, ja generatiivisilla malleilla voidaankin löytää riippuvuuksia luotettavammin esimerkiksi lääketieteen sovelluksissa, joissa käytettävissä on usein vain vähän näytteitä. Väitöskirjassa esitellään myös muihin mallitusperiaatteisiin perustuvia malleja; riippuvien komponenttien analyysi on uusi menetelmä, jolla löydetään monimuotoisempia riippuvuuksia kuin aiemmilla menetelmillä.

Aineistojen välisiä riippuvuuksia voidaan käyttää myös perinteisten ohjaamattomien oppimismenetemien ohjaamiseen. Oppivan metriikan periaate kuvailee, kuinka tutkittavan aineiston ja annetun ohjaussignaalin välisten riippuvuuksien avulla voidaan muodostaa uusi, hyödylliseen informaatioon keskittyvä etäisyysmitta. Tässä työssä parannetaan oppivan metriikan käytössä tarvittavia approksimaatioita ja oppimismenetelmiä.

Contents

Preface

This work has been mainly carried out in the Neural Networks Research Centre and Adaptive Informatics Research Centre of the Laboratory of Computer and Information Science (Department of Information and Computer Science since 2008), Helsinki University of Technology. Part of the work was done in the Department of Computer Science, University of Helsinki, when I was visiting there for a year. The main source of funding has been the Helsinki graduate school in computer science and engineering (HeCSE), and project funding from the Academy of Finland. The work could not have been done without their financial support. I would like to thank PASCAL, an EU network of excellence for Pattern Analysis, Statistical Modeling and Computational Learning, for travel funding, and Tekniikan edistämissäätiö for their personal grant. I am also pleased to having had the opportunity to be a part of the Helsinki Institute for Information Technology (HIIT).

I am grateful to my supervisor Professor Samuel Kaski for teaching me what science is about, and being a constant source of interesting research ideas. His contribution to the thesis is apparent: He is a co-author in all of the publications, and provided valuable comments regarding the Introduction. This thesis would not exist without him.

I would also like to express my gratitude to the reviewers of this thesis, Professors Jukka Corander and Volker Roth, for their expert feedback regarding the thesis.

My sincere compliments belong to all of my other co-authors, Dr. Sinkkonen, Dr. Peltonen, and Mr. Tripathi. Your contributions extend beyond the publications we wrote together.

I would also like to thank the whole MI research group. In particular, thanks to Eika, Jaakko, 2xJanne, 2xJarkko, Leo, and Merja for working all these years beside me. My compliments go also to the whole laboratory, especially to Academician Teuvo Kohonen and Professors Erkki Oja and Olli Simula for providing such a nice and still truly academic working place to work in. I am honored having been a part of it.

I am grateful to my parents and brothers, for providing such a supportive environment to grow up. My thanks also go to all of my friends, in particular to Teemu, Hannu, and Timo. Finally, I am deeply thankful to my wife Mikaela — for everything.

LIST OF PUBLICATIONS

The thesis consists of an introduction and the following publications:

1. Arto Klami and Samuel Kaski. Non-parametric dependent components. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages V-209 – V-212. IEEE, Piscataway, NJ, 2005.

2. Abhishek Tripathi, Arto Klami, and Samuel Kaski. Simple integrative preprocessing preserves what is shared in data sources. *BMC Bioinformatics*, 9:111, 2008.

3. Arto Klami and Samuel Kaski. Generative models that discover dependencies between data sets. In S. McLoone, T. Adali, J. Larsen, and M. Van Hulle, editors, *Machine Learning for Signal Processing XVI*, pages 123–128. IEEE, Piscataway, NJ, 2006.

4. Arto Klami and Samuel Kaski. Probabilistic approach to detecting dependencies between data sets. *Neurocomputing*, 2008. To appear.

5. Arto Klami and Samuel Kaski. Local dependent components. In Zoubin Ghahramani, editor, *Proceedings of the 24th International Conference on Machine Learning*, pages 425–433. Omnipress, Madison, WI, 2007.

6. Jaakko Peltonen, Arto Klami, and Samuel Kaski. Improved learning of Riemannian metrics for exploratory data analysis. *Neural Networks*, 17:1087–1100, 2004.

7. Samuel Kaski, Janne Sinkkonen, and Arto Klami. Discriminative clustering. *Neurocomputing*, 69:18–41, 2005.

SUMMARY OF PUBLICATIONS AND THE AUTHOR'S CONTRIBUTION

In all publications the writing has been a joint effort of all authors, typically in reasonably equal amount. Exceptions are mentioned in the detailed descriptions. The publications are here listed in the order they are presented in the Introduction, which does not correspond to the chronological order.

In Publication 1, a non-parametric extension of canonical correlation analysis (CCA), suited for data not following normal distribution, is presented. The basic approach was jointly developed by the author and S. Kaski, while the author derived the formulas, implemented the method, and carried out all experiments.

In Publication 2, we present a fast and easily applicable data fusion method for finding commonalities between several data sources. The idea and experimental design were jointly developed. The author supervised A. Tripathi in implementing the method and performing the experiments.

In Publication 3, we present a model structure and a basic approach on how to use likelihood maximization for finding statistical dependencies between data sources. Two novel clustering methods are presented. The idea was jointly developed, and the experiments were designed together. The author derived all formulas, implemented the methods, and performed the experiments.

Publication 4 extends the Publication 3. We provide a more general formulation for the generative modeling approach for detecting dependencies, and give a variational Bayesian solution to one of the clustering models presented in the previous publication. The idea and experimental design were joint efforts. The author derived the formulas, implemented the methods, performed the experiments, and wrote most of the publication.

In Publication 5, a Bayesian variant of CCA is introduced. We also introduce an infinite mixture of canonical correlation analyzers. Inference of both methods is made with Gibbs sampling. The author and S. Kaski jointly developed the idea and designed the experiments, while the rest of the work was done by the author, including most of the writing.

Publication 6 contains the most extensive treatment of the learning metrics principle to supervise unsupervised learning methods. Novel approximations and methods are presented and compared. The author participated in designing the experiments, implemented part of the methods, and carried out all experiments.

In Publication 7, we summarize the small sample version of the discriminative clustering (DC) method, and introduce a novel regularization idea based on modeling also the density of the primary data. The author derived formulas for the regularization and implemented part of the methods. The author also participated in designing the experiments, and carried them out.

LIST OF ABBREVIATIONS

AC	Associative clustering
ARD	Automatic relevance determination
BF	Bayes factor
DC	Discriminative clustering
DP	Dirichlet proces
CCA	Canonical correlation analysis
CDA	Confirmatory data analysis
CGH	Comparative genomics hybridization
CRP	Chinese restaurant process
DeCA	Non-parametric dependent components
EDA	Exploratory data analysis
EM	Expectation maximization
GCCA	Generalized canonical correlation analysis
HSIC	Hilbert-Schmidt independence criterion
IB	Information bottleneck
ICA	Independent component analysis
KCCA	Kernel canonical correlation analysis
KL-divergence	Kullback-Leibler divergence
LM	Learning metrics
MDA	Mixture discriminant analysis
MLP	Multilayer perceptron
PCA	Principal component analysis
PLS	Partial least squares
RBF	Radial basis function
RKHS	Reproducing kernel Hilbert space
SDC	Stochastic discriminative clustering
SOM	Self-organizing map
VB	Variational Bayes

LIST OF SYMBOLS

In this thesis boldface symbols are used to denote matrices and vectors. Capital symbols (e.g. $\mathbf{W}$) signify matrices and lowercase symbols ($\mathbf{w}$) column vectors.

D	dimensionality of the data
N	number of observations
$\mathbf{X}, \mathbf{Y}$	data matrices in $\Re^{D \times N}$
$\mathbf{x}$, $\mathbf{y}$	data samples, vectors in $\Re^D$
X, Y	random variables
$p(\mathbf{x})$	probability or probability density of X
$E_{p(\mathbf{x})}[\cdot], E[\cdot]$	expectation over the probability density
$d(\mathbf{x}, \mathbf{y})$	distance between $\mathbf{x}$ and $\mathbf{y}$
ρ	correlation
$I(X, Y)$	mutual information between random variables X and Y
$H(X)$	entropy of random variable X
$\|\cdot\|$	L_2 norm of a matrix or vector
$\{\mathbf{m}_j\}_{j=1}^K$	a collection of K vectors $\mathbf{m}_j$
$\mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$\mathrm{Mult}(N, \boldsymbol{\theta})$	Multinomial distribution with sample size N and parameter vector $\boldsymbol{\theta}$
$\mathrm{Dir}(\boldsymbol{\theta})$	Dirichlet distribution with parameter vector $\boldsymbol{\theta}$
$\mathrm{IG}(\alpha, \beta)$	Inverse Gamma distribution with parameters α and β
$\mathrm{IW}(\mathbf{S}, \nu)$	Inverse Wishart distribution with parameters $\mathbf{S}$ and ν
$\mathrm{DP}(G, \alpha)$	Dirichlet process with base distribution G and concentration parameter α
$\mathbf{J}(\mathbf{x})$	Fisher information matrix evaluated at $\mathbf{x}$
$h(i, j)$	Neighborhood function between indices i and j on a SOM lattice
$K(\mathbf{x}_i, \mathbf{x}_j)$	a kernel function

Chapter 1

Introduction

1.1 General motivation and background

In this thesis, we consider *exploratory analysis* of multivariate data sets. Exploratory data analysis (EDA) is a subfield of *data analysis*, which is a field of study using computational methods to analyze collections of data, such as measurements made from industrial processes, collections of text extracted from the internet, or sensor data from a video surveillance system. EDA considers the task of extracting novel systematic properties from unknown data collections. Typical methods include clustering to group similar measurements together, and visualization of the data collection on a two-dimensional display.

A central subtask in exploratory analysis lies in defining what kinds of properties of the data are relevant. All systematic structures are not interesting; for example, some properties might already be known to exist in the data, or the data may be so complex that instead of studying all aspects of the data, we would want to concentrate on a specific subset of them. Definition of relevance is necessarily tied to the purpose of the analysis, and features or properties that are relevant for one task might not be it for another task.

Most existing methods do not have rigorous solutions for defining relevance, but instead the definition is hidden in the modeling assumptions and choices, such as the selection of a distance measure. For example, an algorithmic clustering method groups similar objects together, and the solution is determined by the form of the similarity measure. Changing the distance measure changes the clustering solution. Typically Euclidean or some other conventional distance measure is used, which makes the choice of the representation for the objects crucial. Application field expertise is needed to choose and preprocess the representations so that relevant results are obtained. For established fields of research, there are often good preprocessing techniques available, but developing those for novel application fields requires significant amount of work.

In this thesis, we consider theory and practical methods for utilizing other data sets to focus the analysis towards the interesting phenomena, aiming to provide more relevant results with less manual work. The idea is to find structure visible not only in the original data, but also in the other data sets with co-occurring observations. This moves up the definition of relevance one level higher in abstraction; instead of needing to specify directly which features reveal the interesting

differences, the user can specify other measurements of the same process.

In this thesis, the basic approach to utilizing other data sets for supervising unsupervised learning is based on *statistical dependencies*. Information shared by two data sets can be found by extracting representations that are mutually informative of each other, i.e. are statistically dependent. This allows defining a novel *data fusion* task for EDA: Combine two data sets by searching for statistical dependencies between them. This can also be regarded as a generalization of noise reduction by averaging over replicate measurements. Averaging as such is applicable only when the measurements share an identical form of representation, but dependencies can be sought between measurements of different dimensionalities. In both cases, we still get more accurate results by ignoring noise specific to individual measurements.

In this thesis, the statistical dependencies between data sets are used to create methods for two different settings. First, two parallel data sets can be analyzed together so that they supervise each other. Neither of the data sets is considered more important than the other, and the task is to find properties shared by both data sets. An example of a useful application would be combining video and audio data to find more relevant information. An unsupervised analysis of a video signal alone would reveal all kinds of moving objects. By supervising the analysis with a paired audio track we can focus on movement related to the audio track, and for example find the face of a person talking. At the same time, we also improve the analysis of the audio track, by being able to ignore sounds that are not related to something shown on the video.

In the other setting, the task is non-symmetric. We wish to analyze only one of the data sets, called the primary data, and the other is only used as a supervision signal. In this thesis, the methods for this setting are based on the *learning metrics principle* (Kaski and Sinkkonen, 2004), which gives a distance measure for the primary data based on the dependencies. Replacing a traditional distance measure with the learning metrics distance turns any unsupervised analysis method into a partly supervised one. An example application could be the analysis of financial data of companies. The learning metrics principle allows focusing the analysis to, for example, information relevant for bankruptcy risk, while still providing useful EDA results, such as illustrations on how changes in profitability or liquidity affect the risk. Note that this differs from the task of *classification*, where we would be interested in predicting the bankruptcy risk for new companies.

In this thesis, we consider the probabilistic approach to extracting and using the dependencies. Some of the models use dependency measures that depend on probability densities, while some define a generative description of the data in the form of a *hierarchical Bayesian model* (see, e.g., Gelman et al. (2003)). The probabilistic modeling framework was chosen due to its rigorous mathematical justification in applications involving uncertainties, such as measurement errors in data collections.

In this introductory part, several potential application fields for the methods are discussed. In the publications, the main application field has been bioinformatics, for three reasons:

1. There are efficient genome-wide measurement techniques that can be used to measure, e.g., the activities of tens of thousands of genes at once. Large data collections are hence measured all the time in this context.

2. The studied processes, typically related to how cells work, are very complex

and to a large extent unknown. There is hence a clear need for EDA.

3. Biologists have worked for decades to create different kinds of annotations for genes and proteins. There exists therefore also additional information that can be used to focus the analysis.

Despite the strong presence of bioinformatics applications in the publications, the methods are more generic. Any application field roughly satisfying the above three criteria could have been considered instead.

1.2 Contributions and organization of the thesis

In this thesis, two main types of contributions are presented. First, we introduce a novel theory and approach to modeling, discussing how dependencies between data sets can be used to focus EDA towards more relevant findings. Second, we present methods for finding the dependencies, and apply them to solving practical problems. In total, eight distinct methods or models for different kinds of tasks are presented. A summary of these methods can be found in Table 7.1 in Chapter 7.

The three main contributions of the thesis can be summarized as:

1. Description of a data fusion scenario where the aim is to find relevant properties shared by two or more data sets with co-occurring samples.

2. Development of theory and methods for finding statistical dependencies between two or more data sources using Bayesian generative models.

3. Development of practical methods for the task of "supervised unsupervised learning", based on the learning metric principle.

The thesis is structured as follows. Chapter 2 is an introduction to modeling and data analysis in general, explaining necessary background for the rest of the thesis. In Chapter 3, measures of statistical dependency and methods for its maximization are presented. A novel dependency-maximizing method is also introduced. In Chapter 4, we explain how dependencies can be used to formulate a data fusion approach for EDA, list potential application fields, and present a novel data fusion method. These two chapters cover the first point on the list of main contributions.

Chapter 5 is about using probabilistic generative models for finding dependencies between data sets. First we discuss how such models can be built in general, and then present concrete models for practical applications. Some of the models are treated in a fully Bayesian fashion. This chapter covers the second main contribution.

In Chapter 6, methods based on the learning metric principle are considered. It covers the third main contribution, discussing mainly practical methods for the task. Finally, Chapter 7 concludes the thesis and lists possible future directions.

Throughout the thesis the ideas and formulations of the novel algorithms are explained, but technical details found in the publications are often not repeated. Presentation of some of the methods has been improved from the original, and the methods are now presented in a unified framework that encompasses all of the methods presented in the publications. In this introduction, the ideas and basic principles of the methods are sometimes graphically illustrated, but numerical results of comparisons present in the publications are not restated.

Chapter 2

Modeling and exploratory data analysis

2.1 Models in data analysis

In data analysis, *models* are used to describe collections of measurements or observations. The purpose is to summarize the process which created the data, so that we can use the model for predicting future observations or understanding the data.

The modeling task can be very roughly divided into three phases: Selecting a collection of possible models, called a *model family*, choosing a criterion for measuring how well a model describes the data, and choosing a specific model that best describes the data that is being analyzed. Possible models are often defined by a set of parameters, the values of which determine the specific model within the family. The choice of the model is often called learning or fitting the model, and, in practice, it means selecting the parameter values so that the chosen criterion of fit, called the *cost function*, is maximized or minimized. All of these choices depend heavily on the exact modeling task, as well as on the properties of the data.

The cost function measures the quality of the model in describing the particular data set. The choice of the cost function is crucial, since it determines in what sense we want to describe the data. In short, the cost function characterizes what kinds of mistakes or deviations in the description are penalized. After choosing the cost function and the model family, the remaining problems are largely computational; how to efficiently find a solution with a good cost function value.

2.1.1 Notation

In this thesis, a *data set* (also called data source) means a collection of N observations or samples, each of which is considered to be an instance of a random variable. The samples are assumed to be independent and follow the same underlying probability distribution, which is assumed to be unknown. In this thesis, the samples are always represented as vectors of real of discrete values. The elements of those vectors are called features, and if we denote by D the number of features then a data set can be represented by a matrix $\mathbf{X}$ that has N columns

and D rows. That is, $\mathbf{X} \in \Re^{D \times N}$. Column vectors $\mathbf{x} \in \Re^D$ are used to denote individual samples, and $x \in \Re$ is used for the special case of univariate samples. When necessary, X will be used to denote the random variable corresponding to $\mathbf{x}$. Probabilities and probability densities are denoted by $p(\mathbf{x})$ or $p(\mathbf{x}|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ contains the parameters of a parametric representation. In most of the cases, the term probability is used to describe both actual probabilities (for discrete $\mathbf{x}$) and probability densities (for continuous $\mathbf{x}$).

Most of the models in this thesis rely on having two or more data sets with *co-occurring samples*. To simplify notation, such models are introduced by using only two data sets, $\mathbf{X}$ and $\mathbf{Y}$. Co-occurrence means that $\mathbf{x}_i$, the ith sample in $\mathbf{X}$, is paired with $\mathbf{y}_i$. That is, they correspond to the same object. A pair of such data sets could equivalently be represented as a single matrix with $D_x + D_y$ features, where D_x denotes the number of features in $\mathbf{X}$ and D_y the number of features in $\mathbf{Y}$. Treating the parts of the matrix separately as individual matrices will, however, often simplify notation and improve the understandability of the models.

All integrals presented in the thesis are definite, and the domain is always over the support of the integrand, unless stated otherwise. For notational simplicity, the domain is not explicitly written, with the understanding that $\int p(\mathbf{x})d\mathbf{x}$ means integration of $p(\mathbf{x})$ over all possible values of $\mathbf{x}$. Typically, there will be no risk of confusion about the domain of $\mathbf{x}$, and usually it will be $\Re^D$.

2.1.2 Generalization and overlearning

A central concept in modeling is the generalization ability of a model. A model learned from a data set should not only describe that particular data set, but also the underlying distribution $p(\mathbf{x})$. This can be studied by measuring whether the model generalizes to new observations that were not available in the learning phase. A model that fits the training data but does not generalize well to new observations is called overfitted or overlearned; it describes not only the process generating the data, but also noise in the training data. A model that describes the underlying probability distribution $p(\mathbf{x})$, however, will generalize well to any new observations following the same distribution.

A large part of the machine learning methodology is devoted to solving the issue of overfitting, since it is present in most learning methods and model families. The less there is data, the more severe the problem becomes, and thus simply obtaining more data would be a handy solution in many cases. This is naturally not always feasible due to possible costs involved in measuring the data, as well as the computational cost of learning. Methods for handling small data sets without overfitting are hence needed. Here, small refers to the number of samples N; the number of features D may still be large, making computational analysis of the data necessary.

Measuring the generalization ability of a model is a non-trivial issue. Given sufficient amount of data, we can leave part of the data out when learning the model, and then measure the performance on the left-out data. This provides a direct estimate on the generalization ability, but the estimate has high variance. A better estimate is obtained with the so-called *cross-validation* procedure (see, e.g., Kohavi (1992)), where the model is learned several times, always using a subset of the data for validation while using the rest for training. There are different variants of cross-validation, such as K-fold cross-validation, where the data is divided into

K subsets out of which K-1 are always used for training, and leave-one-out cross-validation, where all but one sample are used in the training. An alternative approach to measuring generalization ability is based on *bootstrapping* (Efron, 1979), which creates similar but not identical version of the available data set by re-sampling observations.

Generally, a model is more likely to overfit if it is complex, that is, has high effective dimensionality of parameters. A traditional approach to avoiding overfitting utilizes this idea by regularizing the model towards a simpler one. For example, a nonlinear model can be regularized by changing the solution towards a more linear solution, or the parameters of the model can be moved towards zero or other neutral value. Given a way of estimating the generalization ability, be it cross-validation, bootstrapping or something else, we can then choose the parameters of the regularization procedure so that the generalization ability is maximized. Many regularization methods can be implemented by adding a separate regularization term to the cost function.

2.2 Probabilistic modeling and Bayesian analysis

Probabilistic models, also called statistical models, describe the generation of data by probability distributions. Many of the methods described in this thesis follow rather strictly the probabilistic modeling framework, and hence the basic concepts are worth describing here. However, the fundamentals of probability theory or descriptions of different probability distributions are not included. Sufficient background for these can be found for example in (Bernardo and Smith, 1994), but for the majority of the thesis the concepts of probability theory are not needed.

A parametric *generative probabilistic model* defines a probability distribution

$$p(\mathbf{x}|\boldsymbol{\theta}),$$

where $\mathbf{x}$ denotes an observation vector and $\boldsymbol{\theta}$ is a collection of model parameters. The most straightforward approach to learning a model is to find an estimate of the parameters $\boldsymbol{\theta}$, such that the density matches the data available for training, and use the learned parameter values for future prediction. The match can be measured by using the *likelihood function*, which can be stated for a particular data set of conditionally independent (given $\boldsymbol{\theta}$) samples as

$$L(\boldsymbol{\theta}|\mathbf{X}) = \prod_{i=1}^{N} p(\mathbf{x}_i|\boldsymbol{\theta}),$$

the product of probabilities or probability densities of different samples. Maximizing this with respect to $\boldsymbol{\theta}$ gives the model with the highest likelihood of explaining the specific observations we have. In practice, the logarithm of $L(\boldsymbol{\theta}|\mathbf{X})$ is used instead, since the logarithm factorizes the product into a sum of terms, one for each data sample. This does not change the solution, since logarithm is a monotone function.

Probabilistic models can be regularized by assigning *prior distributions* for the parameter values. In short, we assume that the parameter values themselves follow a certain probability distribution. This gives a regularizing effect if the prior distribution prefers parameter values leading to simpler models, or to models that are assumed more likely to be accurate prior to seeing the data $\mathbf{X}$. The objective

in the learning task is then changed from maximizing the likelihood to maximizing the *posterior probability* of the parameters, taking into account both the likelihood of the model and the prior distributions of the parameters. This moves the solution towards the prior, not allowing the model to overfit as much to the data. The less data we have, the stronger the effect of the prior is.

The posterior probability is obtained using the *Bayes' rule* (Bayes, 1763/1958)

$$p(\boldsymbol{\theta}|\mathbf{X}) = \frac{p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{X})}, \tag{2.1}$$

where $p(\boldsymbol{\theta})$ is the prior probability of the parameters, $p(\boldsymbol{\theta}|\mathbf{X})$ is the posterior probability of the parameters given the data, and $p(\mathbf{X}) = \int p(\mathbf{X}, \boldsymbol{\theta}) d\boldsymbol{\theta}$ is the prior predictive probability of $\mathbf{X}$. Note that $p(\mathbf{X})$ is independent of $\boldsymbol{\theta}$, and can thus be ignored when searching for optimal $\boldsymbol{\theta}$. The task of maximizing $p(\boldsymbol{\theta}|\mathbf{X})$ can then be defined as

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \sum_{i}^{N} \log p(\mathbf{x}_i|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}),$$

where the logarithm has been taken to enable factorizing the likelihood over the data points, as well as to separate the prior as an additive penalty term.

The above formulation describes learning as the the task of estimating a single parameter value that fits the data. However, typically we are not interested on the actual parameter values, but instead primarily on the predictions of the model. Better predictions can be made by considering a set of possible models instead of a single one. In *Bayesian analysis* (Gelman et al., 2003), we consider the full *posterior distribution* of the parameters, and change the modeling task from finding a single best model to finding the posterior probability of each of the possible models. In principle, this allows full control over the overfitting issue, since we can use this uncertainty over models in all stages of analysis. If we know the posterior distribution of the parameter values, we can integrate over the possible models to get the expected prediction. In general, if we denote by $f(\boldsymbol{\theta})$ a function that depends on the model, then the Bayesian estimate of $f(\cdot)$ given the observed data is the expectation

$$E_{p(\boldsymbol{\theta}|\mathbf{X})}[f(\boldsymbol{\theta})] = \int p(\boldsymbol{\theta}|\mathbf{X}) f(\boldsymbol{\theta}) d\boldsymbol{\theta}. \tag{2.2}$$

In full Bayesian analysis we cannot ignore $p(\mathbf{X})$, since it is needed to normalize $p(\boldsymbol{\theta}|\mathbf{X})$ to be a probability distribution. Unfortunately, computing $p(\mathbf{X})$ is non-trivial for majority of the interesting models used in real data analysis. This causes severe difficulties in both finding the posterior distribution and computing integrals of the form (2.2), and has lead to extensive literature on approximation methods (see, e.g., Gelman et al. (2003) for a good overview). Some of the methods will be mentioned and briefly discussed in Chapter 5, where they are used for inference in the models developed in this thesis.

It is worth mentioning that while the description of the Bayesian modeling framework here proceeds at a very practical level, describing the prior as a regularization and the posterior distribution as a method for avoiding the need to pick a single solution, there is a rigorous theory behind the framework. Under rather relaxed assumptions, the Bayesian formalism can be considered a direct consequence of being rational in presence of uncertainty (Bernardo and Smith, 1994). Hence, all modeling should in principle be formulated as a Bayesian inference task, conditioned on the prior beliefs of the modeler (which may be highly subjective expert

opinions, or more objective beliefs obtained, for example, though empirical Bayes; see Efron (2005) for a discussion on Bayesian analysis in a bit wider context). In this thesis, the approach is, however, more practical; fully Bayesian analysis is applied only when needed, and modeling approaches based on optimization of costs other than the posterior probability are accepted as reasonable approximations.

2.3 Modeling tasks

A useful viewpoint to modeling tasks is to consider a dichotomy into supervised and unsupervised tasks, based on the goal of the modeling. The categories are not exhaustive, however. In fact, the methods in this thesis borrow from both modeling tasks.

In the following explanations, the citations to example methods have been left out, since the details are not relevant for the rest of the thesis. Readers interested in these general concepts of machine learning and data analysis are asked to consult for example (Bishop, 2006) or (Duda et al., 1999).

2.3.1 Supervised learning

Supervised learning is perhaps the most intuitive of modeling tasks. The task is to learn a mapping from $\mathbf{x}$ to $\mathbf{y}$, given a set of example pairs $\{\mathbf{y}_i, \mathbf{x}_i\}_{i=1}^{N}$. After learning such a mapping, we can predict the value of $\mathbf{y}$ for new samples $\mathbf{x}$. In terms of probabilistic modeling, the task is to learn the conditional distribution $p(\mathbf{y}|\mathbf{x})$, but also formulations based on more general mappings are possible.

Depending on the nature of $\mathbf{y}$, the supervised learning tasks can be divided into two main categories. If $\mathbf{y}$ is categorical, then the task is called classification; we want to predict which category or class a new sample belongs to. The goal can be either the prediction of the whole class distribution, or simply predicting the most likely class for each sample. The latter approach allows deviating quite far from the probabilistic formulation of learning the distribution $p(\mathbf{y}|\mathbf{x})$, by using as a cost function simply the number of incorrect classifications or other more direct measures of classification quality. Example classification methods include K-nearest neighbor classifier, decision trees, and support vector machines.

In the case of continuous $\mathbf{y}$, the task is usually called regression. In regression, $\mathbf{y}$ can be either univariate or multivariate, and the problem is typically to learn some summary statistics, such as the mean and variance, of the conditional distribution. A classical regression method is linear least squares regression, whereas Gaussian process regression is an example of a more advanced regression method.

2.3.2 Unsupervised learning

Unsupervised learning refers to the task of summarizing or modeling data, without considering part of it as a target variable like in supervised learning. In terms of probabilistic modeling, we can characterize unsupervised learning as finding a model to represent the probability distribution $p(\mathbf{x}, \mathbf{y})$. By choosing a model family with easily interpretable parameters, the data can be summarized through the parameter values.

Typical unsupervised learning tasks include clustering, density estimation, visualization, and dimensionality reduction. In clustering, the task is to group sim-

ilar observations together into clusters. This is traditionally performed either by looking for a set of prototypes each describing a group of samples (e.g. K-means clustering and mixture of Gaussians), or by recursive schemes merging or dividing clusters into larger or smaller ones (e.g. agglomerative hierarchical clustering). Spectral clustering methods, in turn, are based on application of linear algebra to matrices of pairwise distances. Many of these different approaches are tightly linked, and it is also possible, for example, to formulate K-means through pairwise distances.

In density estimation, we are directly interested in the density function of the underlying distribution, and estimate that by a parametric approximation $\hat{p}(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})$ or by non-parametric kernel density or nearest neighbor estimators. Dimensionality reduction, in turn, refers to methods that construct a lower-dimensional version of a high-dimensional feature vector $\mathbf{x}$. Dimensionality reduction can be performed, for example, by linear projections (e.g. principal component analysis), by explicitly optimizing a lower-dimensional representation (multidimensional scaling methods), or by various more advanced methods. Special cases of dimensionality reduction, where the dimensionality of the resulting representation is two or three, can be used for visualizing high-dimensional data sets on a screen or paper.

2.3.3 Combining the two basic tasks

Several different approaches that do not fall clearly into either of the two categories above have been introduced. Some of those are discussed here to provide background for the methods of this thesis.

In this thesis, methods for analyzing $\mathbf{x}$ and $\mathbf{y}$ symmetrically, aiming to find what is in common between them, are presented. This task, here coined as *modeling of dependencies*, is fundamentally a special case of unsupervised learning, but it has a close connection to supervised learning as well. The task in modeling of dependencies is to find statistical dependencies between two sets of variables. In practice, this is done by describing the dependencies in a way similar to how the data is described in unsupervised learning. We can look for clusterings such that the cluster indices given by two different clusterings are dependent, or we can find projective transformations such that the lower-dimensional representations are similar. The connection to supervised learning is that we can think of detecting dependencies as a bi-directional supervised learning task; based on $\mathbf{x}$ we should learn to predict $\mathbf{y}$, and vise versa.

Another example of combining supervised and unsupervised learning discussed in this thesis is the so-called *supervised unsupervised learning*. As discussed in Chapter 6, it is actually a special case of the dependency modeling task, but since also a closer connection can be found, it is worth mentioning here separately. The task is to explore $p(\mathbf{x})$ with a presence of a co-occurring $\mathbf{y}$, and the aim is to find properties of $\mathbf{x}$ that are informative of $\mathbf{y}$ as well. The models are familiar unsupervised learning models, but the cost function in the learning takes also $p(\mathbf{y}|\mathbf{x})$, the cost of supervised learning, into account.

Semi-supervised learning (Chapelle et al., 2006) is in a sense opposite to supervised unsupervised learning. In semi-supervised learning, the task is to learn a supervised model and to use unsupervised learning models to help that task. A classical example is a classification setting where we only have the class labels $\mathbf{y}$ for some of the training samples. In a purely supervised setting, the samples $\mathbf{x}$ for

which there are no class labels would be ignored, whereas in semi-supervised learning they are used to improve the classification result. Semi-supervised learning will not be discussed further in this thesis.

2.4 Exploratory data analysis

Of particular interest in this thesis is the task of *exploratory data analysis*, here abbreviated EDA. EDA was named by Tukey (1977), and it considers looking at data in order to suggest novel hypotheses. These hypotheses can be used as a basis for futher data collection, or standard statistical hypothesis testing can be applied to provide further insight on the findings. The hypothesis testing phase is often called *confirmatory data analysis* (CDA), even though it is worth keeping in mind that traditional hypothesis testing cannot formally confirm any hypothesis.

Most unsupervised learning methods are good candidates for the task of EDA, since they provide as a result some kind of summaries. It is, however, also possible to use supervised learning methods for EDA, for example by looking at which features were useful for a certain classification task. This gives a hypothesis that these particular features are related to the classification itself.

EDA is a conceptually difficult task, since the setting directly specifies that there is little prior information on the data. In fully unsupervised EDA, it is difficult to determine which results are useful or correct. In this thesis, partly supervised approaches are considered to make EDA more clearly defined; the supervision is used to focus fundamentally unsupervised methods on more relevant findings. The supervision is provided through searching for statistical dependencies between co-occurring data sets, which means that the supervision only comes into play in defining the data sets, making the approach still feasible for EDA.

Chapter 3

Maximization of statistical dependency

In this chapter, we discuss the concept of statistical dependency and methods that aim at finding dependencies between co-occurring variables. The basic concepts and methods are presented here, whereas applications are discussed in the next chapter.

3.1 Measures of dependency

3.1.1 What is dependency?

In this thesis, the notion of dependency refers to deviation from independency. Independency can be defined either for events or for random variables; in this thesis, the latter case is considered. Independency can be rigorously formulated as a property of the joint probability function $p(\mathbf{x}, \mathbf{y})$: Random variables X and Y are statistically independent if and only if $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y})$ for all $\mathbf{x}$ and $\mathbf{y}$. This property generalizes directly to more than two variables: The joint density factorizes into a product of marginal terms if the variables are independent.

Independency, as such, is thus a very clearly defined and simple concept. In particular, it is essentially binary; either two variables are independent or they are not. Dependency, however, is a continuous quantity. In addition to knowing that two variables are not independent, we typically want to know how strong the dependency is. In this section, several alternative formulations to measuring the strength of dependency are presented.

3.1.2 Mutual information

Mutual information is a concept of information theory, measuring the statistical dependency between two random variables (see, e.g., Cover and Thomas (1991)). For discrete random variables, mutual information is defined as

$$I(X, Y) = \sum_{\mathbf{x}} \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y}) \log \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})p(\mathbf{y})}, \tag{3.1}$$

where the summations go over all possible values of X and Y. In the continuous case the sums are replaced by integrals, and $p(\mathbf{x}, \mathbf{y})$ denotes the joint density function instead of probability, but otherwise the formula is identical.

Mutual information measures the information shared by random variables X and Y, or, alternatively, how much knowing one of the variables reduces the uncertainty about the other. In information theory, the uncertainty can be measured by entropy (Shannon, 1948), and mutual information characterizes the decrease in entropy when knowing one of the variables. Entropy of a discrete variable X is defined as

$$H(X) = -\sum_{\mathbf{x}} p(\mathbf{x}) \log p(\mathbf{x}),$$

giving the mutual information an equivalent formulation as

$$I(X, Y) = H(X) + H(Y) - H(X, Y).$$

If X and Y are independent, that is $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y})$, then $I(X, Y)$ is zero. In the other extreme where the variables are identical, the value reaches the entropy of X (or Y). That is, if you know the value of X then the entropy of Y is reduced to zero. In general, $I(X, Y)$ cannot exceed the entropy of either variable.

An alternative view to mutual information is to consider it as the Kullback-Leibler divergence between two distributions, one assuming that the variables are independent, and the other not. Kullback-Leibler divergence (Kullback and Leibler, 1951) is a measure of discrepancy between two distributions, defined as

$$d_{KL}(p, q) = \sum_{\mathbf{x}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})},$$

and thus mutual information can be written as $d_{KL}\left(p(\mathbf{x}, \mathbf{y}), p(\mathbf{x})p(\mathbf{y})\right)$.

In this work, mutual information is regarded as the standard definition for the strength of statistical dependency. If $p(\mathbf{x}, \mathbf{y})$ was known we could use mutual information to get an exact and accurate characterization of the dependency. In practice, however, we only have finite data sets $\mathbf{X}$ and $\mathbf{Y}$, and hence it is worth considering also measures that can be more reliably estimated from small samples, even though they might not correspond to mutual information.

3.1.3 Correlation

A classical measure of association or dependency between two univariate variables x and y is Pearson's correlation (Galton, 1886; Pearson, 1896), defined as

$$\text{corr}(X, Y) = \rho_{xy} = \frac{E[(x - E[x])(y - E[y])]}{\sqrt{E[(x - E[x])^2]}\sqrt{E[(y - E[y])^2]}}, \tag{3.2}$$

where $E[\cdot]$ denotes the expectation over the joint probability distribution $p(x, y)$. In practice, the expectations are often replaced by population means when estimating the correlation, giving the sample correlation coefficient.

The values of correlation range from -1 to 1. The sign of the measure indicates the nature of the relationship, while the absolute value tells the strength of the dependency. The correlation is 1 or -1 for variables that are linearly dependent, and 0 for statistically independent variables. The converse is generally not true;

the correlation may be zero for non-independent variables, but for the special case of multivariate normal distribution, a zero correlation also implies independence.

In fact, an even stronger connection holds for multivariate normal distribution. It can be shown (see, e.g., Borga (2001)) that for jointly normal x and y (i.e. the concatenation of x and y follows multivariate normal distribution) the correlation and mutual information are related by

$$I(X,Y) = -\frac{1}{2}\log\left(1-\rho_{xy}^2\right).$$

This implies that for jointly normal variables we can use correlation as a dependency measure without loss of generality. For other distributions correlation is not equivalent to mutual information, but should be regarded merely as a measure of linear relationship.

3.1.4 Non-parametric correlation measures

As described in the previous section, Pearson's correlation makes an implicit assumption of multivariate normality of data. Non-parametric measures of correlation have also been presented. These measures are based on ranks of the values, and do not assume any particular type of distribution.

Spearman's rank correlation ρ_{xy}^S (Spearman, 1904) can be regarded as computing the Pearson's correlation between the ranks of the observations on the two variables, but in practice it can be computed directly based on differences between the rankings as

$$\rho_{xy}^S = 1 - \frac{6\sum_i^N d_i^2}{N(N^2-1)}.$$

Here d_i denotes the difference between the ranks of x_i and y_i.

Another rank correlation is Kendall's τ (Kendall, 1938), based on the concept of concordance between sample pairs. A pair of samples is called concordant if they are in the same order in the ranking within both variables, and discordant otherwise. The measure can then be computed as

$$\tau_{xy} = \frac{2(C-D)}{N(N-1)},$$

where C is the number of concordant pairs and D is the number of discordant pairs.

Both Spearman's correlation and Kendall's τ are 0 for uncorrelated variables, and 1 and -1 signify perfect correlation, i.e. identical ranks. While rank correlation measures are applicable to a wider range of distributions than Pearson's correlation, due to their non-parametric nature, they are considerably more difficult to use as an optimization criterion. This is because they are not differentiable, and also due to their higher computational demand.

3.1.5 Bayes factors

Mutual information measures the deviation from independency through Kullback-Leibler divergence between $p(\mathbf{x},\mathbf{y})$ and $p(\mathbf{x})p(\mathbf{y})$, which requires knowing the distributions. In order to estimate mutual information directly based on data, we

will need to estimate the densities. For example Ihler et al. (2004) consider dependency measures based on empirical estimation. However, if we are willing to make some assumptions about the distributions, we often get a better estimate by using a Bayesian measure of fit instead of explicitly estimating the mutual information. For small sample sizes, such an estimate is typically more reliable and accurate than using the mutual information directly, assuming that the distributional assumptions are good enough.

Bayes factor (BF) (Kass and Raftery, 1995) is a ratio used to compare two alternative models, measuring their relative fit. A BF between two models, H_1 and H_0, is given as

$$BF = \frac{p(\mathbf{X}, \mathbf{Y}|H_1)}{p(\mathbf{X}, \mathbf{Y}|H_0)}, \tag{3.3}$$

where $\mathbf{X}$ and $\mathbf{Y}$ denote the data, divided here into two parts for apparent reasons. That is, BF is the ratio of the marginal data densities under the two models. For measuring dependency, we choose H_0 to be a model that makes an independency assumption $p(\mathbf{X}, \mathbf{Y}|H_0) = P(\mathbf{X}|H_0)P(\mathbf{Y}|H_0)$, and H_1 to be a joint model $p(\mathbf{X}, \mathbf{Y}|H_1)$. If the models are otherwise identical, the BF between them becomes a dependency measure.

All the typical difficulties in computing the marginal likelihood apply for BF. In order to be able to compute it, we need to marginalize the parameters $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ of the models out, either analytically or approximatively. For univariate discrete data, this is relatively straightforward by assuming that counts of the pairs of x and y follow a multinomial distribution with a Dirichlet prior. Good (1976) presents practical formulas for different alternative scenarios with different constraints and prior assumptions.

BF is not bounded from above by any simple expression, unlike mutual information and correlation. Nevertheless, it is possible to characterize the strength of dependency quantitatively. Kass and Raftery (1995) provide general guidelines on interpreting the strength of evidence against H_0. They are directly applicable for interpreting the strength of dependency when H_0 adds an independency assumption to H_1.

3.1.6 Kernel-based dependency measures

Correlation measures only linear relationship, but it can be extended to non-linear dependencies by considering the correlation after applying non-linear functions to the observations. Consider an operator $C = \mathrm{corr}(f(\mathbf{x}), g(\mathbf{y}))$, where $f(\cdot)$ and $g(\cdot)$ are arbitrary bounded univariate functions. Rényi (1959) shows that X and Y are independent if and only if C is zero for all bounded functions $f(\cdot)$ and $g(\cdot)$. Intuitively, there cannot be any dependency structure in the data if it cannot be transformed into a space where there would be linear dependency. A non-linear dependency measure, called maximal correlation, can then be defined as

$$\rho_{\max} = \sup_{f,g} \mathrm{corr}\left(f(x), g(y)\right).$$

Taking the supremum over all bounded functions is in general computationally impossible, but in a reproducible kernel Hilbert space (RKHS), which consists of all functions in which a pointwise evaluation is a continuous linear functional, it can be replaced with a computationally feasible kernel-based measure. Gretton

et al. (2005) define the Hilbert Schmidt Independence Criterion (HSIC) as the squared Hilbert-Schmidt norm of the kernelized C. The HSIC can be estimated relatively efficiently, the computational complexity being only N squared. Hence, it is possible to build computational methods relying on optimization of HSIC. Shen et al. (2007) give a variant of independent component analysis (ICA) through minimization of HSIC, Song et al. (2008) formulate clustering though maximization of HSIC between the samples and their cluster assignments, and L.Song et al. (2008) present a dimensionality reduction method that preserves a relation with given auxiliary information, such as class labels.

3.2 Maximization of dependency

A straightforward approach to modeling of dependencies starts from selecting a specific dependency measure and the type of representations used. These fix the modeling task, and the remaining problem is then simply to devise an algorithm to optimize the selected dependency criterion. In this section, several practical methods for modeling dependencies are explained. The methods differ both in the choice of the dependency measure and the form of the representations, but they all still aim to solve the same fundamental task of finding statistical dependencies between the data sets.

3.2.1 Canonical correlation analysis

A classical method for seeking dependencies is the canonical correlation analysis (CCA), originally proposed by Hotelling (1936). For a more recent explanation, see for example (Hardoon et al., 2004). CCA assumes linear projections for representations, and the cost function is the Pearson's correlation.

Let us start with only one-dimensional projections. Then, the cost is to maximize $\mathrm{corr}(S_x, S_y)$, where $\mathbf{S}_x = \mathbf{u}_x^T\mathbf{X}$ and $\mathbf{S}_y = \mathbf{u}_y^T\mathbf{Y}$. Here, $\mathbf{u}_x$ and $\mathbf{u}_y$ are projection vectors to be chosen to maximize the correlation. The norm of the projection vectors is not interesting since it factors out in the correlation (3.2), and thus, we can arbitrarily fix the length to some suitable value. A logical choice is $\mathbf{u}_x^T\mathbf{\Sigma}_{xx}\mathbf{u}_x = 1$ and $\mathbf{u}_y^T\mathbf{\Sigma}_{yy}\mathbf{u}_y = 1$, where $\mathbf{\Sigma}_{xx}$ and $\mathbf{\Sigma}_{yy}$ are the covariance matrices of X and Y, since these constraints allow expressing the covariance in a simpler form. Lagrange's technique for constrained optimization then gives the cost

$$\mathbf{u}_x^T\mathbf{\Sigma}_{xy}\mathbf{u}_y + \frac{\lambda_x}{2}(1 - \mathbf{u}_x^T\mathbf{\Sigma}_{xx}\mathbf{u}_x) + \frac{\lambda_y}{2}(1 - \mathbf{u}_y^T\mathbf{\Sigma}_{yy}\mathbf{u}_y),$$

where $\mathbf{\Sigma}_{xy}$ is the cross-covariance of X and Y, and λ_x and λ_y are Lagrange multipliers.

Algebraic manipulation of the derivatives of the above equation shows that both Lagrange multipliers are equal to the maximal correlation between S_x and S_y, i.e. $\lambda_x = \lambda_y = \rho = \mathbf{u}_x^T\mathbf{\Sigma}_{xy}\mathbf{u}_y$. This allows expressing the optimization problem as

$$\mathbf{\Sigma}_{xy}\mathbf{\Sigma}_{yy}^{-1}\mathbf{\Sigma}_{yx}\mathbf{u}_x = \rho^2\mathbf{\Sigma}_{xx}\mathbf{u}_x$$
$$\mathbf{\Sigma}_{yx}\mathbf{\Sigma}_{xx}^{-1}\mathbf{\Sigma}_{xy}\mathbf{u}_y = \rho^2\mathbf{\Sigma}_{yy}\mathbf{u}_y.$$

Solving these generalized eigenvalue problems gives a pair of projections $\mathbf{u}_x$ and $\mathbf{u}_y$, together with the maximal correlation ρ.

After finding the projections with the highest correlation, we should look for other projections that have maximal correlation, with the restriction that they are orthogonal to the previous ones. In the case of CCA, the constraint is that the projected values of different components should be uncorrelated. If we denote the ith projection vector of the $\mathbf{x}$-space by $\mathbf{u}_x^{(i)}$, then $(\mathbf{u}_x^{(i)})^T \mathbf{X}\mathbf{X}^T \mathbf{u}_x^{(j)} = 0$ for all $i \neq j$. This can be equivalently be written as $(\mathbf{u}_x^{(i)})^T \mathbf{\Sigma}_{xx} \mathbf{u}_x^{(j)} = 0$. Matrices $\mathbf{U}_x$ and $\mathbf{U}_y$ are used to denote the whole set of projections.

In practice, we can find all projections at once by solving a generalized eigenvalue problem $\mathbf{AU} = \rho\mathbf{BU}$, where $\mathbf{U}$ denotes the concatenation of $\mathbf{U}_x$ and $\mathbf{U}_y$, and

$$\mathbf{A} = \begin{pmatrix} 0 & \mathbf{\Sigma}_{xy} \\ \mathbf{\Sigma}_{yx} & 0 \end{pmatrix}, \tag{3.4}$$

$$\mathbf{B} = \begin{pmatrix} \mathbf{\Sigma}_{xx} & 0 \\ 0 & \mathbf{\Sigma}_{yy} \end{pmatrix}. \tag{3.5}$$

Solving this equation is straightforward (see (Melzer, 2002) for discussion on relative merits of the possible solution strategies), and it gives a unique global optimum. CCA is hence fast to compute (linear in N and cubic in D_x and D_y) and easy to use, which makes it a feasible choice for many applications. However, the linearity and the implicit assumption of global normality, which follows from using correlation as the cost function, are often too rigid constraints. It is also worth noticing that CCA can only be applied in scenarios where the number of samples N is considerably larger than the number of features D. If D was larger than N, then there would always be perfect correlations due to the features not being linearly independent, and, in practice, CCA already overfits severely if D is close to N.

Extensions and generalizations of CCA

Classical CCA can be extended to more than two data sets in various ways (Kettenring, 1971), each retaining some but not all properties of the ordinary two-space CCA. Perhaps the most intuitive extension, called here the generalized CCA (GCCA), is described for example in (Bach and Jordan, 2002). It is solved using a generalized eigenvalue problem analogous to CCA, i.e. $\mathbf{AU} = \lambda\mathbf{BU}$, where also $\mathbf{A}$ and $\mathbf{B}$ follow the same structure. That is, $\mathbf{B}$ is a block-diagonal matrix of the covariances of the data sets, and $\mathbf{A}$ is the covariance of the concatenation of all data sets minus $\mathbf{B}$. The generalization retains the connection to mutual information (termed multi-information in case of several variables) in case of multivariate normal data, and again a unique global optimum can be found, but the method has no interpretation in terms of correlation, a measure defined only for two variables.

Other natural extensions of CCA are obtained by keeping the cost function equal to the correlation, but changing the type of the projections. Sigg et al. (2007) introduce a non-negative and sparse CCA, where the projection matrices should have only a few positive elements while the rest are zero. The sparseness is achieved by L_1 regularization, while the non-negativeness is handled by adding constraints to the optimization problem, formulated as iterated regression instead of directly solving the eigenvalue problem. These constraints improve the interpretability of CCA in applications where non-negativity is a desired property.

Several methods extending CCA to non-linear projections have been presented. Becker and Hinton (1992) formulate a self-taught neural network for discovering

surfaces in random-dot stereograms, although they do not explicitly mention CCA. Instead of correlation, their method optimizes a different criterion that is still equivalent to mutual information for Gaussian data. Becker (1996) discusses other variants of the same basic principle. Both Hsieh (2000) and Lai and Fyfe (1999) make the connection to CCA more explicit, by introducing non-linear CCA using multilayer perceptrons (MLP) instead of linear projections. These methods have the advantage of allowing non-linear mappings, but on the other hand they are far more difficult to optimize than classical CCA. In particular, the property of having a unique global optimum is lost.

Kumar et al. (2002) use radial basis function (RBF) networks to obtain non-linear CCA by first making a non-linear transformation for the observations and then using classical CCA on the transformed variables. This is closely related to the kernel CCA (Fyfe and Lai, 2000; Akaho, 2001; Bach and Jordan, 2002), where the RKHS theory is used to enable non-linear mappings through kernel functions. A kernel $K(\mathbf{x}_i, \mathbf{x}_j)$ over all pairs of samples can be used to efficiently compute inner products between $f(\mathbf{x}_i)$ and $f(\mathbf{x}_j)$, so that the choice of the kernel implicitly defines a function $f(\cdot)$ that maps the data samples into some typically high-dimensional feature space. Despite the non-linearity, linear algebra still gives unique global optimum for kernel CCA. While kernel CCA enables more complex mappings, it has also a few downsides. First, the choice of kernel defines the mapping $f(\cdot)$ only implicitly, and typically it is not possible to interpret the results in terms of the original variables, causing difficulties in interpretation. Also, if using flexible kernels that map the data implicitly into an infinite-dimensional feature space, the solution needs to be regularized heavily. Otherwise the algorithm will always find perfect correlations due to the fact that the dimensionality is higher than the number of data points. Choosing the regularization as a function of the number of samples is discussed in (Fukumizu et al., 2007).

3.2.2 Non-parametric dependent components

In this section, a novel generalization of CCA, introduced in Publication 1, is presented. It extends CCA into data not following the normal distribution, by replacing the correlation as a cost function with an empirical estimate of mutual information. The method is still based on linear projections, and applicable to the same setting as CCA.

For multivariate normal data, Pearson's correlation is equivalent to mutual information, and hence CCA finds all potential dependencies. For non-normal data, however, a better measure of dependency would be desirable. The non-parametric correlation measures explained in Section 3.1.4 would be natural alternative candidates, but they are not particularly suited for optimization due to being based on the ranks of the samples. This makes them non-differentiable, and also rather computationally complex. Using Spearman's correlation as optimization criterion for one-dimensional projections is discussed in (Dehon et al., 2000), where projection pursuit strategy is used to find a single component at a time, with the additional restriction that the set of possible projection vectors is limited to be finite.

A computationally more feasible approach is to consider a numerical approximation of the mutual information as the cost function. If the approximation is chosen such that the cost remains computationally tractable and differentiable, then the cost can be explicitly maximized by conventional optimization strategies.

Publication 1 presents a novel method applying this strategy, here called DeCA for *dependent component analysis*. The presentation shown here differs noticeably from the one in the original publication, aiming to be more understandable and following the basic framework used in the thesis.

The mutual information (3.1) has two properties that make it difficult as a measure for continuous variables: It involves joint probability densities, and an integral over the whole probability space. To enable optimization we need to approximate both of these. In DeCA, the densities are estimated using Parzen kernel estimates (Parzen, 1962)

$$\hat{p}(\mathbf{x}_i) = \sum_{j \neq i} \mathrm{N}(\mathbf{x}_i | \mathbf{x}_j, \mathbf{\Sigma}),$$

where $\mathrm{N}(\cdot|\boldsymbol{\mu}, \mathbf{\Sigma})$ denotes the density of the normal distribution with mean $\boldsymbol{\mu}$ and covariance $\mathbf{\Sigma}$, and the summation is over all other data points. The difficult integration task is, in turn, solved by using a plug-in strategy where the integral over a probability distribution is replaced by a sum over samples from that distribution. Together with the density estimates, this leads to the approximation

$$I(S_x, S_y) = \int\int p(\mathbf{s}_x, \mathbf{s}_y) \log \frac{p(\mathbf{s}_x, \mathbf{s}_y)}{p(\mathbf{s}_x)p(\mathbf{s}_y)} d\mathbf{s}_x d\mathbf{s}_y \approx \sum_{i=1}^{N} \log \frac{\hat{p}(\mathbf{s}_x^i, \mathbf{s}_y^i)}{\hat{p}(\mathbf{s}_x^i)\hat{p}(\mathbf{s}_y^i)},$$

where $\mathbf{s}_x^i$ and $\mathbf{s}_y^i$ denote the projected variables associated with the ith sample pair, and S_x and S_y denote the corresponding random variables. The estimator is consistent (see Beirlant et al. (1997) and Paninski (2003) for discussion on estimating entropy and mutual information), though not necessarily particularly accurate for small sample sizes. Note that the original presentation in Publication 1 starts from a rather different kind of formulation, namely the Bayes factor between hypotheses of dependent and independent distributions, but the actual cost function is still the same. The original publication mentions the asymptotic connection to mutual information, but does not show it explicitly.

As the model family consists of linear projections, we have $\mathbf{S}_x = \mathbf{W}_x^T \mathbf{X}$ where $\mathbf{W}_x$ is a projection matrix. Here, $\mathbf{S}_x$ is a matrix consisting of the projections $\mathbf{s}_x$ of individual samples, and analogously for $\mathbf{S}_y$. If $\mathbf{W}_x$ has D_x colums, then the density estimation is performed on a space that has $D_x + D_y$ dimensions. Density estimation in high-dimensional spaces is difficult, and thus two simplifications are made in DeCA. First, the method is optimized one component at a time, thus needing to estimate the density only in a two-dimensional space, and the covariance matrix of the Gaussian distributions used in the Parzen estimate is restricted to be diagonal.

Given the approximation for the cost function, the task is simply to maximize it with respect to the projections $\mathbf{w}_x$ and $\mathbf{w}_y$. In DeCA this is done with the conjugate gradient method, but in principle any gradient-based optimization method could be used. After finding the first component, the task is to find the next component so that the projected variables are independent of each other. Unfortunately, that is in practice difficult, and hence an approximative solution is proposed in Publication 1: The contribution explained by the first component is removed from data before searching for another component. This can be obtained by the deflation procedure

$$\tilde{\mathbf{X}} = \mathbf{X} - \mathbf{w}_x \mathbf{w}_x^T \mathbf{X},$$

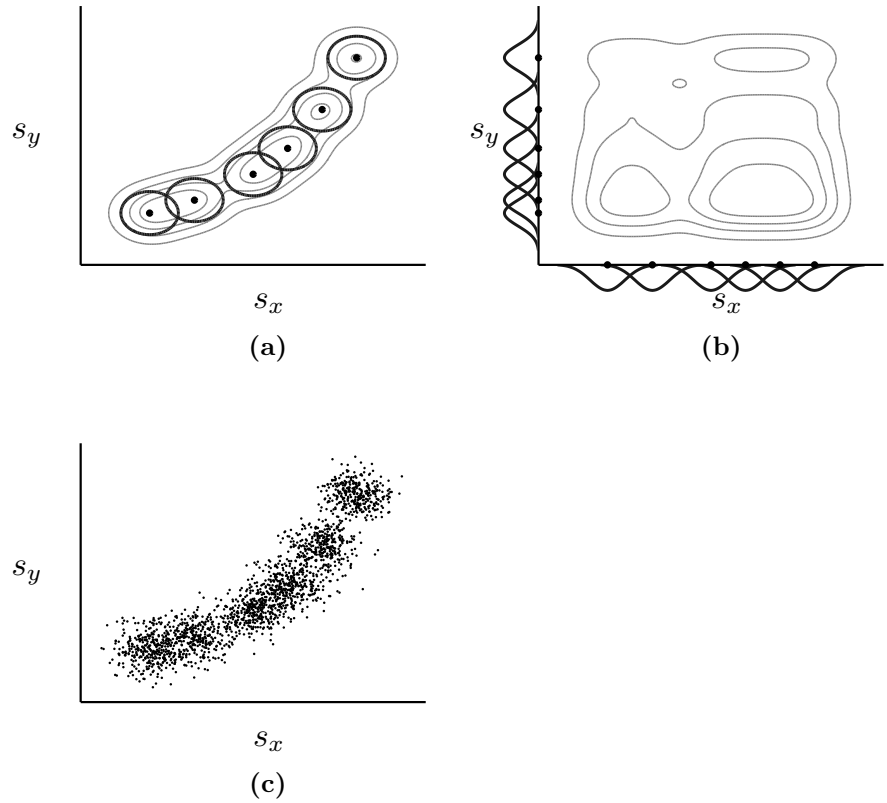

Figure 3.1: An illustration of the two density estimates used in DeCA. In subfigure **(a)** the density is estimated in the joint projection space, whereas in subfigure **(b)** the joint density is assumed to factorize as a product of two marginal densities. The joint estimate matches the projected data $(\mathbf{s}_x, \mathbf{s}_y)$, shown in subfigure **(c)**, clearly more accurately. The illustration hence shows an example of a case where the dependency between the two variables is high. DeCA tries to find this kind of projection spaces. Note that for illustrational purposes the density is here estimated by just 6 component densities, instead of the Parzen density estimate used in DeCA. Light gray lines denote contours of the full density estimate, and dark thick lines depict the component densities.

and analogously for $\mathbf{Y}$. The second component is then estimated by applying DeCA to $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$. Alternatively, the orthogonality criterion of CCA (that is, the projected variables are uncorrelated) could be used, which is achieved by the deflation

$$\tilde{\mathbf{X}} = \mathbf{X}\left(\mathbf{I} - \frac{1}{\|\mathbf{w}_x^T\mathbf{X}\|^2}\mathbf{X}^T\mathbf{w}_x\mathbf{w}_x^T\mathbf{X}\right).$$

This approximates more closely the requirement of having independent projections.

The idea of the method is illustrated graphically in Figure 3.1, where two density estimates are shown. High dependency is achieved if the estimate not assuming dependency (i.e. modeling directly $p(\mathbf{s}_x, \mathbf{s}_y)$) is more accurate than the one assuming dependency (modeling $p(\mathbf{s}_x)p(\mathbf{s}_y)$), and the model aims at finding projections for which this difference is maximized. The figures illustrate a case where the difference is clear, as can be seen from the poor estimate obtained with the independence assumption.

Fisher and Darrell (2004) present a related method. The model is formulated

as a hypothesis test: Find such projections that the hypothesis of the two latent signals being dependent is more likely than the hypothesis of them being independent. The test is formulated as a log-ratio, which is in turn re-interpreted as mutual information. The mutual information is then approximated as

$$\hat{I}(S_x, S_y) = \hat{H}(S_x) + \hat{H}(S_y) - \hat{H}(S_x, S_y),$$

where each entropy term is approximated using a separate second-order Taylor-series, based on Parzen density estimates. Even though the original cost function, as well as the density estimation technique, are the same, the resulting optimization problem is different since the approximations used are different.

We became recently aware of another similar method by Yin (2004). Like DeCA, the method maximizes a non-parametric estimate of mutual information in the projection space with Gaussian kernels. The technical details of the methods are very similar, including the same cost function, despite having been developed independently. Compared to (Yin, 2004), the Publication 1 includes the connection to the Bayes factor and empirical tests with much larger data sets, including a comparison to kernel CCA. It also treats the case of more than two data sets, by replacing mutual information with multi-information. Yin (2004), in turn, includes tests for choosing how many significant components the data supports and discusses the consistency of the method.

3.2.3 Associative clustering

All the previous methods have been examples of projection methods. Another common model family is based on clustering. In unsupervised learning in general, clustering has been an extremely widely studied task, but for the dependency modeling task it is somewhat more complicated and therefore not as widely applied. In our framework, changing from projections to clustering is in principle simple, but the discrete cluster assignments mean that the approximations for the mutual information and the optimization procedures are quite different.

In this section, a clustering algorithm for finding mutual dependencies is reviewed. Associative clustering (Kaski et al., 2005) (AC) is a simple clustering model which, in a sense, extends K-means clustering into a case where the task is to maximize dependencies between two clusterings. For a sample pair $(\mathbf{x}, \mathbf{y})$, two cluster indices s_x and s_y are defined separately as

$$\begin{aligned} s_x = k \quad &\text{iff} \quad \|\mathbf{x} - \mathbf{m}_x^k\|^2 < \|\mathbf{x} - \mathbf{m}_x^j\|^2 \quad \text{for all} \quad j \neq k, \\ s_y = k \quad &\text{iff} \quad \|\mathbf{y} - \mathbf{m}_y^k\|^2 < \|\mathbf{y} - \mathbf{m}_y^j\|^2 \quad \text{for all} \quad j \neq k, \end{aligned}$$

where $\mathbf{m}_x^j$ denotes the jth cluster prototype in the $\mathbf{x}$-space and similarly for $\mathbf{y}$-space. In other words, each sample is assigned to the cluster with the closest prototype, separately for the $\mathbf{x}$- and $\mathbf{y}$-spaces.

Since the cluster indices are discrete, it is beneficial to consider an alternative form of representation that helps in estimating the dependency between S_x and S_y. A *contingency table* is a representation where the counts of samples having a certain combination of s_x and s_y are collected as a two-dimensional table. Given such a table, normalized to sum to one, we could estimate mutual information by a simple summation directly using (3.1). However, for finite data, we can improve the accuracy by using the Bayes factor (3.3) instead. Here, the model H_1 assumes

that the contents of the table are generated by a multinomial with an independent parameter for each element, whereas H_0 assumes that the parameters are formed by a product of parameters for the marginals of the table. Following Good (1976), we can integrate the parameters out, leading to

$$BF = \frac{\prod_{i,j} \Gamma(n_{ij} + n^0)}{\prod_i \Gamma(n_{i\cdot} + n^0) \prod_j \Gamma(n_{\cdot j} + n^0)} \tag{3.6}$$

as the final measure. Here, n_{ij} is the number of samples in the (i, j)th cell in the contingency table (that is, samples that have ith cluster in the $\mathbf{x}$-space and jth cluster in the $\mathbf{y}$-space), $n_{\cdot j}$ and $n_{i\cdot}$ denote the column and row sums, and n^0 is a prior parameter. A high value of BF denotes dependency and a low value signals independency.

The AC method is formulated as finding the cluster prototypes $\{\mathbf{m}_x^j\}_{j=1}^{K_x}$ and $\{\mathbf{m}_y^j\}_{j=1}^{K_y}$ so that the Bayes factor of the induced contingency table is maximized. This is, unfortunately, computationally difficult due to the discrete nature of the contingency table; the cost is not differentiable. In AC, this is overcome by introducing a smoothed version of the contingency table. Specifying soft membership functions for the clusters allows placing fractions of the samples into the clusters. This makes the counts, and hence also the BF, differentiable with respect to the centroids $\mathbf{m}_x$ and $\mathbf{m}_y$. The logarithm of (3.6) where the n are replaced with the soft sums is then maximized with conjugate gradients. Again any other gradient-based optimization method could be applied instead.

3.2.4 Symmetric information bottleneck

In the discrete domain, a family of methods has been developed around the concept of information bottleneck (IB) (Tishby et al., 1999). The basic IB, discussed in more detail in Chapter 6, is a directed model, aiming to cluster a discrete variable so that the cluster assignment would be informative of another discrete variable.

In (Friedman et al., 2001), a symmetric variant of IB is presented, among other variants, making the concept relevant also for dependency maximization. In symmetric IB the task is to cluster two variables, here denoted by X and Y, so that the cluster indices of the $\mathbf{x}$-space, S_x, are informative of Y, and vise versa. As usual in the IB framework, the complexity of these clusterings is controlled by requiring that the mutual information between X and S_x is minimized (and similarly for Y). This creates the bottleneck in the name of the method, forcing X to be compressed into S_x. Despite the formulation as two separate IB tasks, the actual method can be interpreted also as maximizing the mutual information between S_x and S_y. The final task is then simply to maximize the mutual information between the cluster indices, while minimizing the mutual information between the cluster indices and the original variables, separately for each space.

The symmetric IB can be solved by an iterative algorithm closely resembling the algorithms for the traditional IB. The solution is controlled by a Lagrange parameter λ. Varying its value creates a path of solutions, so that increasing the value gives solutions of increasing complexity.

Another relevant extension of the IB principle is the Gaussian IB, presented in (Chechik et al., 2005). Instead of discrete variables, the method works on variables assumed to be normally distributed and resembles closely the canonical correlation analysis. The method finds the same subspace as CCA, but the tradeoff parameter

can be used to control the dimensionality of the subspace, while also determining the scale of the projection vectors. It is, however, worth noting that the Gaussian IB is not inherently symmetric, but instead it finds the projection only for $\mathbf{X}$. It still finds for $\mathbf{X}$ the same projection vectors as CCA does, even though the projection for $\mathbf{Y}$ is restricted to be a unit matrix.

3.3 Sidenote: Minimization of dependency

In this thesis, only methods maximizing dependencies are considered. That is, the task is always to find mappings that capture dependencies as well as possible. It is, however, worth mentioning also the extensive literature and methodology devoted to doing the opposite, i.e. looking for representations that minimize the dependencies.

Minimization of dependencies is used particularly in the task of blind source separation, where the aim is to detect true source signals from an observed mixture in a setting where both the sources and the mixing are unknown. Methods such as the independent component analysis (ICA) (Hyvärinen et al., 2001) solve this by assuming that the true underlying signals are statistically independent, hence trying to find an inverse mixing that would result to maximally independent sources. Often the statistical dependency is measured by some indirect manner, such as measuring the non-Gaussianity of the signals by higher moments. However, also methods working directly on approximations of mutual information have been presented (Van Hulle, 2008), as well as methods that aim to minimize the HSIC criterion (Shen et al., 2007). Some of the kernel-based variants of ICA solve the problem through kernel-CCA (Fyfe and Lai, 2000; Bach and Jordan, 2002).

Chapter 4

Dependencies and data fusion

In the previous chapter, methods for finding maximally dependent representations for two data sets with co-occurring samples were presented. In this chapter, potential uses for such methods are discussed. The focus is on novel application scenarios in the field of data analysis and machine learning. In particular, we present how statistical dependencies between data sets can be used for data fusion in exploratory data analysis.

4.1 Data fusion by searching for dependencies

Data fusion is a subfield of data analysis that aims at combining several data sources in order to improve the accuracy of analysis. Data fusion, also known as sensor fusion, has been widely applied especially in classification tasks and in supervised learning in general (Hall and Llinas, 1997), where measuring the accuracy is straightforward and the task is clearly defined. Solving the data fusion task is then, in principle, easy; simply use all sources to the extent that they can improve the classification accuracy. Practical difficulties naturally remain. Data fusion in supervised learning will not be discussed further in this thesis.

In this thesis, we focus on data fusion in unsupervised learning. This means that our goal still remains at finding systematic regularities from a collection of measurements, using as an input a collection of data sets with co-occurring samples. The simplest approach would be to ignore the fact that the features are divided into separate data sets, and instead consider the concatenation of all observations. Any unsupervised learning method can be applied on such a data set. However, the information on the sources being of separate origin is not explicitly used in such an approach, which means that a potential source of information is ignored.

Recently, a wide range of unsupervised methods utilizing the natural split into several data sources have been presented. A term *multi-view learning* is collectively used to describe methods searching for consensus between the views, using various criteria to define the consensus. The underlying assumption is that if models learned based on different views agree with each other, they are likely to generalize better. Various multi-view approaches, including also supervised methods, are

described in proceedings of the ICML 2005 Workshop on 'Learning with Multiple Views' (Rüping and Scheffer, 2005). Here we consider a related approach, where the consensus is defined through statistical dependencies between the data sources. This approach uses the information on source identities to focus the analysis on properties only revealed in a data fusion setting.

Perhaps the most intuitive reason for looking for dependencies between two data sets is noise reduction. Practically all observations contain noise, i.e. some stochastic variation for which we cannot know the instantaneous value. Such noise can be caused for example by the measurement process itself, and it is often assumed to be independent between the samples. If we have two sets of measurements of the same samples, then we can assume that the noise is independent also between the data sets. In such a setting, looking for dependencies between the data sets should be a principled way of removing noise. This can be seen as a generalization of averaging over several measurements; in a dependency-maximization framework, the averaging is simply replaced by a more general mapping of the features.

Another, somewhat more philosophical, reason for looking for dependencies between data sets is to use dependencies as a definition for what is interesting in the data. Completely unsupervised models describe everything in the data collection, within the limits of the selected model family, but the dependency modeling framework can be used for more directed analysis. If we define the task as searching for information that is present in all of the data sets, then looking for dependencies is a good solution. In particular, it changes the task so that we will find information that could only accidentally be revealed by looking at any of the sources alone; the primary variation in those may or may not be related to the commonalities. Notice that here the approach differs slightly from the traditional multi-view learning setting. In multi-view learning it is typically assumed that each source alone is sufficient for learning a good model, whereas here a model learned based on any one source alone could be considerably different or even misleading.

While all the methods presented in Chapter 3 find dependencies between the data sets, it is not necessarily straightforward to use all of them in this kind of a data fusion. The methods provide separate mappings for $\mathbf{X}$ and $\mathbf{Y}$, whereas for data fusion it would often be better to have a single representation. The separate representations can, however, in many cases be combined. Next we present a justified way of combining the outputs of CCA, and in Chapter 5 novel methods that are directly formulated though a single shared representation are introduced.

4.1.1 CCA-based preprocessing for data fusion

In Publication 2, we show how the separate representations that CCA gives for $\mathbf{X}$ and $\mathbf{Y}$ can be combined to obtain a data fusion solution. This turns CCA into a practical data fusion tool that is fast and easily applicable, and also acts as a demonstration of how also more complex data fusion methods can be built on the idea of dependency maximization.

The method relies on the alternative interpretation of CCA as a two-step procedure. First, each data set is preprocessed separately to remove all within-data variation, and then the main variation remaining in the collection of all data sets is extracted. This process is illustrated in Figure 4.1. The fundamental goal in the first step is to remove all variation that could be extracted by the model used

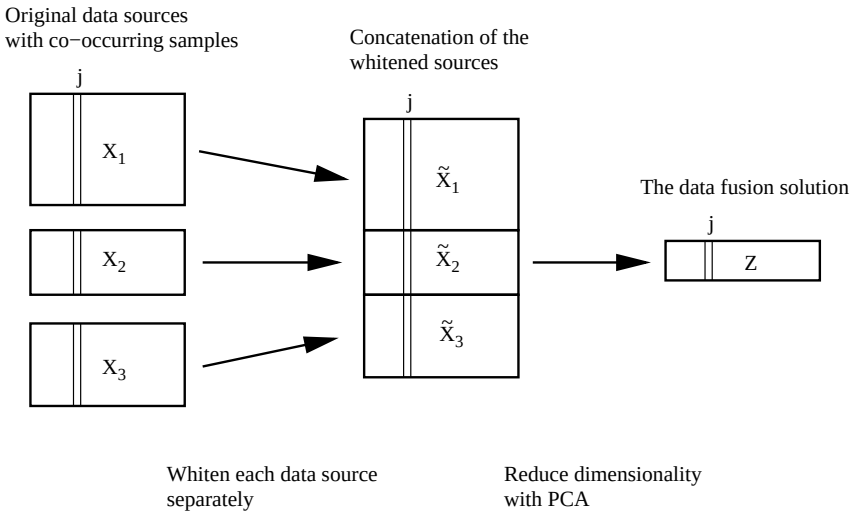

Figure 4.1: An illustration of the data fusion process that aims at finding linear dependencies in the data collection. In this example, three data sources with different dimensionalities are given, and the task is to find what these sources have in common. First, each source is whitened to remove within-data variation, then the whitened sources are concatenated, and finally principal component analysis (PCA) is used to create a lower-dimensional representation that only captures the shared effects. The rectangles represent data matrices with columns as samples and rows as features. The jth sample has been marked to demonstrate the co-occurrence of the samples in the data sets.

in the second step, so that if the data collection has no dependencies between the data sets, then the second step cannot extract anything. Any structure found by the second step must then be dependencies.

In Publication 2, the second step is performed simply by the principal component analysis (PCA) (Hotelling, 1933) of the columnwise concatenation of preprocessed data sets. PCA is a classical method that searches for linear projections of maximal variance. The preprocessing step matching this is a traditional procedure called *whitening*, which is a linear transformation that leads to the preprocessed data having a unit covariance matrix. In practice, the whitening can be performed as

$$\tilde{\mathbf{X}} = \mathbf{\Sigma}_x^{-1/2}\mathbf{X},$$

where $\mathbf{\Sigma}_x$ denotes the covariance of X. PCA cannot find any structure in $\tilde{\mathbf{X}}$, but instead finds that any projection direction has unit variance.

From a concatenation of non-independent whitened data sources, the PCA step is able to extract directions that have higher variance, which must be a result of dependencies between the different preprocessed data sets. Independent linear preprocessing steps cannot create such dependencies, and thus they must already exist in the original data sets. The remaining task is then to simply pick a suitable dimensionality for the PCA projection, so that only dimensions with structure are kept. In Publication 2, a test based on randomization is proposed. A set of random data collections satisfying the independency assumption are sampled from the normal distribution, and the variances of the projections of the real data are compared to the ones obtained when applying the algorithm to the randomized data sets.

Despite the two-step procedure, the method is actually a re-description of classical CCA. Details of the connection can be found in Publication 2. The final data fusion solution $\mathbf{Z}$ can be written as

$$\mathbf{Z} = \mathbf{U}_x\mathbf{X} + \mathbf{U}_y\mathbf{Y},$$

where $\mathbf{U}_x$ and $\mathbf{U}_y$ are the CCA projection matrices of the chosen dimensionality. The result extends to more than two data sets by using the generalized CCA. In other words, the fused data set is simply the sum of the canonical scores (the projected values). That shows how generalized CCA can be used to combine co-occurring data sets into a single representation. The result can then be used as a source for any unsupervised or supervised method needed for further analysis.

4.1.2 The residual variation

In typical modeling tasks, the residual of the model is considered pure noise. In the data fusion approach based on detecting dependencies, it may, however, make sense to also analyze the variation remaining in the data collection after extracting the dependencies. Natural data sets may have very complex information present in some of the data sources, and in some applications, analyzing also such residual variation could be of use.

In fact, we could even take the analysis of the residual variation in one of the sources as the main modeling task. Instead of studying the dependencies we could use the dependency modeling approach to remove variation deemed uninteresting, due to it being visible in another source that can be analyzed more accurately by other methods.

For the methods presented in Chapter 3, the residual variation can typically be extracted as a post-processing step. For example, the variation extracted by CCA can be removed from a certain data set by a linear transformation that subtracts the variation explained by the the projections. In Chapter 5, methods that have explicit representation for both the dependencies and the data set–specific variation are presented, allowing at least in principle selecting directly which of the latent information sources are deemed relevant.

4.2 Applications

Since the introduction of the canonical correlation analysis (Hotelling, 1936), methods aiming at capturing dependencies between two (or more) variables have been extensively used in statistics. Typically, the tasks have been confirmatory in nature; the outcome of the analysis has been, for example, whether two variables correlate or not, or how strong a correlation can be found between two sets of variables. In this work, however, we focus on dependency maximization tasks in exploratory data analysis, and briefly review recent applications A common feature to many of these applications is that we are also interested in the actual latent variables between which the dependency is sought, not just in the degree of dependency or in the parameters of the mappings.

4.2.1 Bioinformatics

Bioinformatics has been an active application field for EDA methods during the recent years. Since the late 1990s, efficient measurement techniques, such as DNA

microarrays, have made it possible to make vast numbers of simultaneous measurements. Today, it is already commonplace to be able to measure the level of expression (i.e. the abundance of messenger RNA) jointly for all human genes. As biological systems are also very complex and typically not fully understood, the relatively inexpensive measurement techniques have made bioinformatics a field where EDA is seriously needed. Data fusion is also particularly important in bioinformatics, since we can get several kinds of measurements from the same system, yet none of them necessarily measures exactly the interesting process directly. Looking for dependencies between several sources might help in focusing the analysis, since it allows ignoring potentially very significant variations caused by complex processes not related to the property under examination.

Kaski et al. (2005) use associative clustering to study orthologous genes (i.e. genes that have common evolutionary origin) of human and mouse. The clustering solution finds regularities and irregularities in the function of the genes, revealing information on conservation of the gene functions in the evolutionary process. The results show, for example, that the dependency structure between the two organisms comes mainly from cell maintenance tasks which are critical for survival. Also genes that differ considerably in their function between the two species are found.

Nikkilä et al. (2005) study environmental stress response in yeast by searching for shared response based on a collection of stressful treatments. Environmental stress response refers to response to perturbations from the normal growth conditions, such as changing the temperature or the concentration of oxides in the environment. Most changes also cause effects specific to that particular change, and thus looking for dependencies between data sets measured under different stressful conditions should reveal the stress response as the variation shared between the conditions. Nikkilä et al. (2005) use both CCA and associative clustering to detect the dependencies. First, dependencies between expression measurements of stressful conditions are sought with CCA, and then AC is applied to find dependency structure between the results of the CCA step and a transcription factor binding data set. The latter step finds explanations for the phenomena detected by the first one. The same example, without the latter step, is used in Publication 2.

Microarrays can be used to measure not only the expression of genes but also changes in the genome itself. Comparative genomic hybridization (CGH) refers to methods measuring the relative amount of DNA sequences, and reveals amplifications and deletions of parts of the chromosome. Chromosomal changes are frequent, for example, in cancers. Combining both expression and amplification measures is thus helpful in analysis of cancer responses (Berger et al., 2006). If we assume that commonalities between amplification and expression measurements in cancer patients are caused by the cancer, then dependencies between these measurements should help in detecting the fingerprint of cancer.

Other examples of the application of CCA or its extensions to microarray data include Nymark et al. (2007) considering time series of expression in asbestos-exposed cell lines, and Parkhomenko et al. (2007) discussing use of sparse CCA to correlate expression with genotypes. In metabolomics, the study of chemical compounds in cells, partial least squares (PLS), which seeks directed dependencies between two multivariate data sets, is widely used (Steinfath et al., 2008). While methods searching for symmetric relations are not so commonly applied, there are also recent metabolomics studies involving the use of CCA (Meyer et al., 2007).

Another active field of study in bioinformatics is proteomics, where the objects of interest are proteins, not genes. Of particular interest are the interactions between proteins, such as the formation of protein complexes that function together, and proteins taking part in signal transduction. Yamanishi et al. (2004) use kernel CCA in predicting protein-protein interactions based on a collection of different data sources. The KCCA is here used as a kind of preprocessing method: It gives a representation where a classification task (whether two proteins interact or not) is easier to solve. Yamanishi et al. (2003) present another application of kernel CCA to bioinformatics. The KCCA is used to simply find the dependencies between more than two sources that do not have vectorial representations, extending classical CCA analysis to novel types of data.

4.2.2 Multimodal content analysis

Another natural application field for dependency-seeking methods is the analysis of digital content that comes in multiple domains. For example, a video sequence not only contains the video stream but in most cases also sound, and occasionally also attached textual content, such as subtitles. There is a natural pairing (time) between all of these signals, and finding dependencies between them can be used to find parts of one domain that correspond to parts in another.

Fisher and Darrell (2004) and Sigg et al. (2007) study the task of detecting speakers from a video. By searching for dependencies between the audio and video tracks, they are able to detect which parts of the video signal correspond to the speech on the audio track. This enables localizing the mouth of the speaker, since the movement around that area must correlate with the audio, thus improving detection of which one of the potentially many humans in the video is talking, even in cases with many simultaneous speakers. Fisher and Darrell (2004) use a method based on non-parametric estimation of mutual information, while Sigg et al. (2007) apply a version of classical CCA that is restricted to find sparse non-negative projections, providing better interpretability of the components.

Another classical example of multimodal content is images and their captions. Often the caption explains something that is visible in the image, and thus searching for dependent representations of both the captions and the images should find text and image features focused on describing the actual content of the images, instead of variation that is irrelevant in the context of the captions. For example, in a collection of pictures of animals, we would like to have features that separate the different animals from each other but are invariant to the lighting conditions of the images, or even to whether the image is a photograph or a cartoon illustration. Searching for dependencies with the captions, which typically do not describe the lighting conditions, should help in that task. In (Farquhar et al., 2006; Hardoon et al., 2004), this kind of approach is used for the annotation of images, using kernel CCA for finding the dependencies.

It is also possible to search for dependencies between texts written in two languages. Li and Shawe-Taylor (2006) study a cross-language information retrieval (IR) and a document classification task by using kernel CCA as a preprocessing before the IR or classification algorithm. The dependencies between the two languages give features that are more informative about the content compared to typical features based on the frequencies of words, resulting in increased accuracy in the tasks.

4.2.3 Other applications

Recently, there has been some work on applying dependency-seeking methods on the analysis of functional brain imaging. Functional magnetic resonance imaging (fMRI) can be used to measure the activity of different brain regions at sufficiently high frequency, making it hence in principle possible to infer which parts of the brain are active at different tasks. Traditionally, this is studied by designing experiments where a test subject is performing a certain isolated task at given times. The activity pattern at that time should then correspond to the task at hand.

For more natural stimuli it is, however, difficult to create good experimental designs. For example, a person watching a movie is simultaneously using his visual and auditory cortices, as well as having brain activity not related to the movie. It is no longer possible to simply examine the activities, but instead it is necessary to try to infer which of the activities is caused by which aspect of the stimuli. Ylipaavalniemi et al. (2007) apply DeCA, described in Publication 1, to such a task in order to find the dependencies between the stimuli and the brain activities preprocessed by independent component analysis (ICA). Also, for example, Hardoon et al. (2007) and Friman et al. (2001) consider applications of CCA to fMRI data.

Dependency maximization methods have also been used in climatology. In addition to the numerous applications of classical CCA, also some more advanced methods have been used. Fern et al. (2005) build a mixture of CCAs to study correlations between precipitation and vegetation index on global scale, and Hsieh (2001) uses a non-linear variant of CCA.

Chapter 5

Generative approach to dependency modeling

In this chapter, theoretical concepts related to using Bayesian generative models to find statistical dependencies are introduced, and practical methods stemming from the theory are presented. The theory is largely a novel contribution of the thesis.

5.1 Why generative approach?

As explained in Chapter 2, the problem of overfitting can often be solved with the Bayesian approach. The Bayesian modeling theory explains how the modeling should be done in principle, and the remaining problems are mainly computational and application-related; what the model family should be, and how the posterior distribution of the parameters can be effectively approximated. Of particular interest are generative models that describe the distribution of the observations.

However, for dependency modeling tasks, there are no existing established Bayesian generative models. This is fundamentally caused by the fact that the cost function of dependency modeling is not formulated as a likelihood, but instead as the correlation, mutual information, or some other measure of statistical dependency. There are naturally several approaches to tackling the overfitting issue with other methods, mainly different kinds of regularization methods (Vinod, 1976) and sampling-based model complexity estimation (Yin (2004), Publication 2), but a generative approach has been lacking. Considering the widely recognized status of generative models in other modeling tasks, it would be interesting to be able to apply them for detecting dependencies as well.

To further motivate the approach, the practical advantages of generative Bayesian modeling can be summarized as:

1. **Measurable characterization of relative quality:** While any cost function can be optimized to find the best solution, most cost functions, including mutual information in the dependency maximization setting, do not provide a natural way of characterizing how much worse another solution with a different cost function value is. Likelihood, instead, has the advantage that a difference

in the log (posterior) density directly tells how much more likely a certain solution is, given the observed data.

2. **Justified treatment of uncertainties:** From the first point it follows directly that uncertainties can be modeled in a justified way. As we can quantify the differences in probability, we can define a full distribution over possible solutions. This allows marginalizing predictions over the whole distribution, instead of needing to select a single model.

3. **More explicit modeling assumptions:** In Chapter 3, the fundamental task in dependency modeling was defined as finding mappings that have maximal mutual information, and specifying a particular approach involves choosing an approximation for the mutual information. The choice of the approximation is a rather implicit assumption, which sometimes makes it difficult to understand what a certain algorithm is doing and what kind of conditions it requires in order to work appropriately. The generative approach makes the modeling assumptions more explicit, making it easier to check how well the assumptions hold for a certain data or task. Explicit modeling assumptions also make it easier to change the assumptions if needed, resulting in novel methods for situations where the existing ones do not work well.

4. **Possibility to include as parts of bigger models:** The probabilistic modeling framework allows building hierarchical models or networks consisting of smaller probabilistic models, while still being able to estimate the whole model jointly in a justified way. In a non-probabilistic setting, we would typically need to optimize each part separately and then combine the parts using a possibly heuristic method, or alternatively derive complex optimization algorithms along the lines of the back-propagation idea used in neural network optimization (Rumelhart et al., 1986). While this may work in some situations, it is an advantage to be able to optimize the full model using the same basic principle in optimization and inference throughout the model. Note, however, that computational issues may still make learning of hierarchical Bayesian models difficult in practice.

We can also motivate the need for a generative approach to dependency modeling through the success of generative modeling in other related tasks. Generative and non-generative approaches have earlier been compared for conditional models and supervised learning. A classification task can be solved either by directly maximizing a classification criterion based on $p(y|\mathbf{x})$, or by constructing a generative model $p(y, \mathbf{x})$ and deriving $p(y|\mathbf{x})$ via Bayes' rule (2.1). Rubinstein and Hastie (1997) compare these approaches, called discriminative and informative learning, empirically, leading to a conclusion that if the generative model matches the underlying distribution, then the latter approach is generally better. Another comparison is given in (Ng and Jordan, 2002), where the generative approach is shown to give better results on small data sets, while the discriminative approach outperforms it on large data sets.

For symmetric dependency maximization tasks, no such comparisons have been possible, since the generative alternatives of the methods have been lacking. Given the theory and models presented in this thesis, it is possible to study whether similar observations generalize to the symmetric case. In Publication 5, this is briefly studied for one particular model, and the result is that the findings of Ng

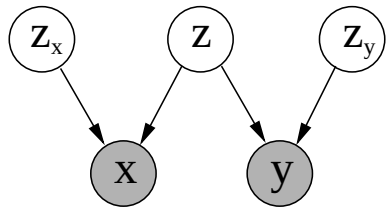

Figure 5.1: A graphical model representation of a general latent variable model structure for detecting statistical dependencies with generative models. Here, $\mathbf{z}$ denotes a latent source common to both observed variables, $\mathbf{x}$ and $\mathbf{y}$, whereas $\mathbf{z}_x$ and $\mathbf{z}_y$ denote latent sources specific to the observed variables. Gray shading indicates observed variables.

and Jordan (2002) do seem to generalize; the generative approach leads to more accurate results on small data sets.

5.2 Detecting dependencies with generative models

For building generative dependency maximization models we use latent variable models (Bishop, 1999a). Latent variable models are a super-family of models that include unobserved variables paired with individual data samples. For example, a clustering model can be formulated as a latent variable model where we have a latent variable z_i for each sample $\mathbf{x}_i$, and z_i indicates which cluster the sample belongs to.

If the task is to learn what the data sets have in common, it makes sense to assume that there is an unknown, latent, source that is shared by all of them. In addition, we need to allow all data sets to have variation that is not shared with the other data sets. This can be conveniently modeled with other latent sources, separate for each of the data sets.

Figure 5.1 shows a generative model structure following the assumptions above. It is a conventional probabilistic model, so all tools of standard Bayesian machinery are directly applicable. However, the link to the dependency maximization task is at this stage merely intuitive, since there is no particular reason to assume that the posterior distribution of the shared latent source would necessarily correspond to the solution which we would find by explicitly maximizing the dependency. It can, however, be shown that under certain assumptions this is the case.

If the task is to find the shared signal $\mathbf{z}$, we can consider the data set–specific latent signals, $\mathbf{z}_x$ and $\mathbf{z}_y$, as nuisance parameters; we are not interested in them as such, and thus marginalize them out. This is not, however, enough to guarantee that the shared signal would capture (only) the dependencies. Instead, we need to make one important assumption: The part of the model that is specific to each data set needs to be an appropriate one. That is, it needs to be flexible enough to model all the data set–specific variation. This is a common assumption made in Bayesian modeling. In many modeling tasks, deviations from the assumption are often not that problematic, since we still get as a result the best possible approximation of the true distribution that is within the selected model family. However, in the case of dependency modeling, the choice is more critical; while we still may get a good approximative model, incorrect distributional assumptions for the data set–

specific parts will cause the shared latent signal to also model variation specific to the data sets. This is empirically demonstrated in Publications 3 and 4.

It is worth noting that the formulation used here is particularly suitable for the data fusion idea presented in Chapter 4. Most traditional dependency-seeking methods are formulated through separate representations for $\mathbf{x}$ and $\mathbf{y}$, but here the model assumes a single shared latent source $\mathbf{z}$. The latent source can be directly used as a new fused representation for the data in applications where we want to extract the dependencies as a preprocessing step for further analysis. The explicit representations $\mathbf{z}_x$ and $\mathbf{z}_y$ for the data set–specific latent sources may also be useful in practical analysis.

5.2.1 General approach

In this section, a general approach to building generative latent variable models for detecting dependencies is presented, following Publication 4. It is assumed that the variation in a data set can be decomposed into a part described by the shared latent source and to another that is specific to that particular data set, and that the actual observation is generated by adding up these two effects. This can be formulated as

$$\begin{aligned}\mathbf{x} &= \mathbf{f}(\mathbf{z}|\mathbf{W}_x) + \mathbf{g}(\mathbf{z}_x|\mathbf{B}_x) + \epsilon_x,\\ \mathbf{y} &= \mathbf{f}(\mathbf{z}|\mathbf{W}_y) + \mathbf{g}(\mathbf{z}_y|\mathbf{B}_y) + \epsilon_y,\end{aligned}$$

where $\mathbf{f}(\cdot|\mathbf{W}_x)$ and $\mathbf{g}(\cdot|\mathbf{B}_x)$ denote arbitrary functions parameterized by $\mathbf{W}_x$ and $\mathbf{B}_x$ (and analogously for $\mathbf{y}$), and ϵ_x and ϵ_y denote noise that is independent between samples and unstructured (i.e. follows some simple distribution).

In a dependency exploration task, we want to find the posterior distribution of $\mathbf{z}$, given the observed data. It can be either $p(\mathbf{z}|\mathbf{x})$, $p(\mathbf{z}|\mathbf{y})$, or $p(\mathbf{z}|\mathbf{x},\mathbf{y})$, depending on the specific inference task. In all cases, we need to marginalize over $\mathbf{z}_x$ and $\mathbf{z}_y$. In the simplest case, we can consider the maximum likelihood estimation of the actual parameters, leading to the marginalization task

$$p(\mathbf{z},\mathbf{x},\mathbf{y}|\hat{\mathbf{W}},\hat{\mathbf{B}}) = \int\int p(\mathbf{z},\mathbf{z}_x,\mathbf{z}_y,\mathbf{x},\mathbf{y}|\hat{\mathbf{W}},\hat{\mathbf{B}})d\mathbf{z}_x d\mathbf{z}_y,$$

where $\hat{\mathbf{W}}$ and $\hat{\mathbf{B}}$ denote estimates of $\mathbf{W}$ and $\mathbf{B}$. The Bayes' rule can then be used to find $p(\mathbf{z}|\mathbf{x},\mathbf{y},\hat{\mathbf{W}},\hat{B})$ or any of the other posteriors of interest. Performing such integration would be difficult in general, but here we have two simplifications that make it computationally feasible. We assumed that $\mathbf{x}$ and $\mathbf{y}$ are conditionally independent given $\mathbf{z}$, and that $p(\mathbf{x}|\mathbf{z},\mathbf{z}_x)$ is an additive composition of terms depending only on $\mathbf{z}$ and $\mathbf{z}_x$ separately. These lead to the simplified expression

$$p(\mathbf{z},\mathbf{x},\mathbf{y}|\hat{\mathbf{W}},\hat{\mathbf{B}}) = p(\mathbf{z})\left(\int p(\mathbf{z}_x)p(\mathbf{x}|\mathbf{z},\mathbf{z}_x)d\mathbf{z}_x\right)\left(\int p(\mathbf{z}_y)p(\mathbf{y}|\mathbf{z},\mathbf{z}_y)d\mathbf{z}_y\right), \quad (5.1)$$

where conditioning on $\hat{\mathbf{W}}$ and $\hat{\mathbf{B}}$ has been left out for brevity on the right hand side. The expression involves separate integrals over $\mathbf{z}_x$ and $\mathbf{z}_y$, and as $\mathbf{z}$ is fixed inside the integrals, the difficulty of the marginalization task only depends on $\mathbf{g}(\mathbf{z}_x|\mathbf{B}_x)$ and ϵ_x (and analogously for $\mathbf{y}$).

5.2.2 On marginalization

The basic approach presented above relies on the ability to perform the marginalization task in (5.1). In all of the models presented in this chapter, this marginalization is done analytically, leading to an exact marginal distribution. This is possible due to two rather restricting assumptions, namely that $\mathbf{z}_x$ and $\mathbf{z}_y$ follow normal distribution, and that $\mathbf{g}(\mathbf{z}_x|\mathbf{B}_x)$ is a linear transformation of the latent variable. For most other choices, it would not be possible to marginalize $\mathbf{z}_x$ and $\mathbf{z}_y$ out in a closed form.

There should be no theoretical problems in extending the approach to cases where the marginalization can be approximated with sufficient accuracy, using for example variational approximation or Markov chain sampling methods (see, e.g. Gelman et al. (2003)). The issue of sufficiency is in this thesis left as an open question. Further work would be needed to answer questions like "How will approximative marginalization change the ability to capture only the dependencies in $\mathbf{z}$?" and "Is it possible to somehow correct the possible deviations caused by approximative marginalization?"

Also worth noting is that the marginalization of $\mathbf{z}_x$ and $\mathbf{z}_y$ is here regarded as a necessary step to guarantee detecting the dependencies. However, as explained in Section 4.1.2, we might in some applications be interested also in the data set–specific latent signals. Hence, modeling also those explicitly would be preferable. Again, the possibility of this is left as an open question. As a practical solution we consider solving the data set–specific signals with a post-processing step after finding the dependencies. In general, first $\mathbf{z}$ is inferred by marginalizing $\mathbf{z}_x$ and $\mathbf{z}_y$ out, and then the data set–specific structure is found by modeling $\mathbf{x}$ and $\mathbf{y}$ separately given a fixed (distribution for) $\mathbf{z}$.

5.3 Probabilistic canonical correlation analysis

Bach and Jordan (2005) interpret canonical correlation analysis as a probabilistic generative latent variable model. In their work, a model structure and a set of distributional assumptions are proposed, and the connection to CCA is proven for the maximum likelihood solution of the model. Earlier also Bie and Moor (2003) considered a similar interpretation. Here, however, an explanation following Publications 3 and 4, is given.

The model structure of Figure 5.1 is abstract, and in essence only specifies certain independencies. To specify an actual model, we also need to give (conditional) probability distributions for the variables. We start with the latent variables, which are *a priori* assumed to be uninformative. Here we choose

$$\mathbf{z}, \mathbf{z}_x, \mathbf{z}_y \sim \mathrm{N}(\mathbf{0}, \mathbf{I}).$$

That is, we assume all three latent variables to follow the multivariate normal distribution with zero mean and unit covariance. The dimensionalities of these three variables may differ, and throughout the description the dimensionalities are not explicitly mentioned. As CCA is a linear projection method, we should have the observed data depend on a linear transformation of the latent variables. With

additive Gaussian noise, we get

$$\mathbf{x}|\mathbf{z}, \mathbf{z}_x \sim \mathrm{N}(\boldsymbol{\mu}_x + \mathbf{W}_x\mathbf{z} + \mathbf{B}_x\mathbf{z}_x, \sigma_x^2\mathbf{I}),$$
$$\mathbf{y}|\mathbf{z}, \mathbf{z}_y \sim \mathrm{N}(\boldsymbol{\mu}_y + \mathbf{W}_y\mathbf{z} + \mathbf{B}_y\mathbf{z}_y, \sigma_y^2\mathbf{I}),$$

where $\mathbf{W}_x$, $\mathbf{W}_y$, $\mathbf{B}_x$, and $\mathbf{B}_y$ are matrices of suitable dimensionality, and σ_x^2 and σ_y^2 are scalar variance parameters. The mean parameters $\boldsymbol{\mu}_x$ and $\boldsymbol{\mu}_y$ are considered as parts of the linear mappings $\mathbf{f}(\mathbf{z}_x|\mathbf{W}_x)$ and $\mathbf{f}(\mathbf{z}_y|\mathbf{W}_y)$.

Following the guidelines of Section 5.2.1, we need to make sure that the data set–specific parts can model all variation inside a single data set. Given the assumption of multivariate normality, this corresponds to requiring the dimensionality of $\mathbf{z}_x$ and $\mathbf{z}_y$ to match the dimensionality of $\mathbf{x}$ and $\mathbf{y}$, respectively. This also allows us to fix both variance parameters σ_x^2 and σ_y^2 to zero, since the data can be completely modeled with the latent sources. Marginalization over $\mathbf{z}_x$ and $\mathbf{z}_y$ can then be performed analytically, resulting in

$$\mathbf{z} \sim \mathrm{N}(\mathbf{0}, \mathbf{I}),$$
$$\mathbf{x}|\mathbf{z} \sim \mathrm{N}(\boldsymbol{\mu}_x + \mathbf{W}_x\mathbf{z}, \mathbf{B}_x\mathbf{B}_x^T),$$
$$\mathbf{y}|\mathbf{z} \sim \mathrm{N}(\boldsymbol{\mu}_y + \mathbf{W}_y\mathbf{z}, \mathbf{B}_y\mathbf{B}_y^T).$$

Here $\mathbf{B}_x\mathbf{B}_x^T$ can equivalently be parameterized as $\boldsymbol{\Psi}_x$, and analogously for $\mathbf{y}$, since a sum of D_x outer products of linearly independent vectors spans the space of $D_x \times D_x$ positive definite matrices. The dimensionality of $\mathbf{z}$ is a free parameter, controlling the dimensionality of the shared latent source.

The above model is exactly the probabilistic model for CCA as given by Bach and Jordan (2005). Theorem 2 in that publication states that for the maximum likelihood solution of the model, we have

$$\hat{\mathbf{W}}_x = \boldsymbol{\Sigma}_{xx}\mathbf{U}_x\mathbf{M}_x,$$
$$\hat{\mathbf{W}}_y = \boldsymbol{\Sigma}_{yy}\mathbf{U}_y\mathbf{M}_y,$$
$$\hat{\boldsymbol{\Psi}}_x = \boldsymbol{\Sigma}_{xx} - \hat{\mathbf{W}}_x\hat{\mathbf{W}}_x^T,$$
$$\hat{\boldsymbol{\Psi}}_y = \boldsymbol{\Sigma}_{yy} - \hat{\mathbf{W}}_y\hat{\mathbf{W}}_y^T,$$

where $\mathbf{U}_x$ and $\mathbf{U}_y$ denote the CCA projection matrices. $\mathbf{M}_x$ and $\mathbf{M}_y$ are arbitrary matrices with spectral norms smaller than one, such that $\mathbf{M}_x\mathbf{M}_y^T = \mathbf{P}$, a diagonal matrix having the canonical correlations on its diagonal. The proof can be found in the original publication. A more intuitive interpretation is that regardless of $\mathbf{M}_x$ and $\mathbf{M}_y$, the expectations $E[\mathbf{z}|\mathbf{x}]$ and $E[\mathbf{z}|\mathbf{y}]$ lie in the subspace that corresponds to the space spanned by the first CCA projections, up to the dimensionality of $\mathbf{z}$.

The probabilistic interpretation of CCA has been used or studied further in a number of publications. Archambeau et al. (2006) present an extension that replaces Gaussian noise with t-distributed noise to make the model more robust to outliers, whereas Leen and Fyfe (2006) and Fyfe and Leen (2006) consider formulations involving Gaussian processes (Rasmussen and Williams, 2006). Archambeau et al. (2006) also present a method for solving the rotational ambiguity caused by $\mathbf{M}_x$ and $\mathbf{M}_y$. This makes it possible to find also the actual CCA projections, instead of just the subspace.

5.4 Bayesian canonical correlation analysis

The probabilistic interpretation of CCA is theoretically interesting, but there is often little need to find the solution by maximizing the likelihood of the model. In practice, directly solving the eigenproblem described in Chapter 3 is usually faster and guaranteed to converge to the global optimum. However, the probabilistic interpretation opens up interesting possibilities, such as extensions by making different probabilistic assumptions or constructing mixture models, and a fully Bayesian treatment of the model. Here, we first consider the Bayesian treatment introduced in Publication 5, and in Section 5.5.1 we provide one extension, a clustering model for searching dependencies (Publication 4). Both of these are novel contributions.

As described in Chapter 2, the Bayesian treatment involves finding the posterior distribution of model parameters given the observed data. This provides an inherent "regularization", so that predictions can be made averaging over the whole posterior instead of using a single, most likely overfitted, model. In the following, we present suitable prior distributions to complement the probabilistic CCA into a fully generative model, and discuss how the posterior distribution, which is analytically intractable, can be approximated.

For computational convenience, conditionally conjugate priors (see e.g. Gelman et al. (2003)) are used for all parameters. In practice, we specify the prior for the covariance matrices $\boldsymbol{\Psi}_x$ and $\boldsymbol{\Psi}_y$ to be inverse Wishart, and the prior for the mean $\boldsymbol{\mu} = [\boldsymbol{\mu}_x; \boldsymbol{\mu}_y]$ to be normal. For the projection matrices $\mathbf{W} = [\mathbf{W}_x; \mathbf{W}_y]$, there are a few possibilities, depending on the independence assumptions made in the prior. We adopted the automatic relevance determination (ARD) prior, which has been previously used in for example the Bayesian principal component analysis (Bishop, 1999b). In ARD prior the elements of $\mathbf{W}$ are drawn from zero-mean normal distribution, with a hierarchical inverse Gamma prior for the variances of the columns. The advantage of the ARD prior is that it can, in a sense, automatically select the number of components by pushing variances of unnecessary columns towards zero. In practice, we can hence use $\mathbf{z}$ of full dimensionality (minimum of the data dimensionalities), and afterwards count the number of non-zero columns. However, it is worth noticing that in a Gibbs sampling approach the columns will not be driven exactly to zero, and therefore slight post-processing is needed for identifying the actual dimensionality.

The independence assumptions are summarized as a graphical model in Figure 5.2. The probabilistic description of the model is then given as

$$\begin{aligned}
\beta_i &\sim \mathrm{IG}(\alpha_0, \beta_0), & \mathbf{w}_i &\sim \mathrm{N}(0, \beta_i \mathbf{I}), \\
\boldsymbol{\Psi}_x &\sim \mathrm{IW}(\mathbf{S}_0^x, \nu_0^x), & \boldsymbol{\Psi}_y &\sim \mathrm{IW}(\mathbf{S}_0^y, \nu_0^y), \\
\boldsymbol{\mu} &\sim \mathrm{N}(0, \sigma_0^2 \mathbf{I}), & \mathbf{z} &\sim \mathrm{N}(0, \mathbf{I}), \\
\mathbf{x} &\sim \mathrm{N}(\boldsymbol{\mu}_x + \mathbf{W}_x \mathbf{z}, \boldsymbol{\Psi}_x), & \mathbf{y} &\sim \mathrm{N}(\boldsymbol{\mu}_y + \mathbf{W}_y \mathbf{z}, \boldsymbol{\Psi}_y),
\end{aligned} \tag{5.2}$$

where IW denotes the inverse Wishart distribution and IG is the inverse Gamma distribution. $\mathbf{W}$ is the row-wise concatenation of $\mathbf{W}_x$ and $\mathbf{W}_y$, $\boldsymbol{\mu}$ is the concatenation of $\boldsymbol{\mu}_x$ and $\boldsymbol{\mu}_y$, and the variables with subscript 0 are prior parameters. The parameters β_i control the variances of the columns $\mathbf{w}_i$ of the projection matrix. For brevity, conditioning on the variables on the right hand side is not explicitly written.

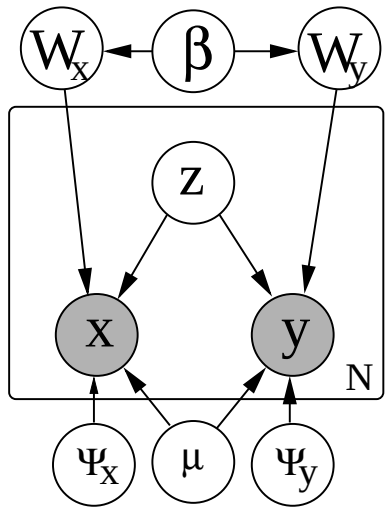

Figure 5.2: A graphical model representation of the generative model for Bayesian CCA. Observed nodes are shown with gray background, and the plate indicates repetition over N samples.

For approximating $p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{Y})$, where $\boldsymbol{\theta}$ includes all the parameters and latent variables of the model, we use Gibbs sampling (Geman and Geman, 1984; Gelfand and Smith, 1990). In (Wang, 2007), a variational Bayes (VB) approach (Jordan et al., 1999) was independently derived for almost exactly the same model, the only difference being different β_i parameters for $\mathbf{W}_x$ and $\mathbf{W}_y$ instead of a shared parameter like in the model (5.2). The variational approximation has the advantage of giving an easily interpretable parametric approximation of the posterior, whereas the sampling approach only provides samples from the posterior. However, Gibbs sampling gives, at least in principle, samples from the true posterior, whereas the variational approximation makes additional independence assumptions and hence does not represent the true posterior. Typically the choice between these two options would depend on the application and practical limitations. For maximal accuracy, the Gibbs sampling approach is often better, but the variational approximation is typically computationally less intensive.

Applying Gibbs sampling to the model (5.2) is relatively straightforward. The Gibbs sampling proceeds by updating each of the parameters at a time, drawing a new value for it from the conditional posterior given the data and the current values of all other parameters. At the limit of infinite number of iterations, this is guaranteed to converge to a process that produces random samples from the true posterior distribution (see e.g. Gelman et al. (2003)). All distributions in the model have conditionally conjugate priors (the prior is conjugate given fixed values for other parameters), and thus the conditional posteriors are relatively easy to derive; they always follow the same type of distribution as the prior, and hence the task is merely to find the equations for the parameters. The actual sampling formulas are given in Publication 5, including the formulas needed to infer $\mathbf{z}_x$ and $\mathbf{z}_y$ given $\mathbf{z}$.

5.4.1 Local dependent components

One severe restriction of CCA in practical applications is that it assumes a global linear dependency over all samples in the data sets. This rarely holds for real data sets. Making the same assumption locally, so that it holds only for a subset of samples, would be feasible in a wider set of applications. Such locality assumption can be easily made in the generative modeling framework: We can construct a hierarchical model that has canonical correlation analyzers as submodels. In this section, a novel model presented in Publication 5 is explained.

A traditional way of changing a global modeling assumption into a set of local assumptions is to build a *mixture model* (see McLachlan and Peel (2000) for an extensive text-book account). A mixture model is an additive model where component densities are summed together to form the full model. An intuitive way to interpret the generative process is that first a latent variable indicating which component to use is drawn from a discrete distribution, and then the actual observation is drawn from the component density. This is fundamentally a description of a clustering process, with potentially complex structure within each cluster. Hence, for the remainder of this section the term 'cluster' is used to denote the component densities, to avoid confusion with the use of the term 'component' to indicate linear projection.

We could relatively easily derive formulas for optimizing a mixture of CCAs by maximizing the likelihood with an expectation maximization (EM) algorithm (Dempster et al., 1977). However, here we continue with the Bayesian approach and thus extend the Gibbs sampling to the mixture case. Furthermore, to avoid determining the number of clusters used in the mixture, a non-parametric Bayesian approach of *infinite mixture models* (Rasmussen, 2000) is adopted. The infinite mixture approach makes the model more truly local, in the sense that increasing the number of samples allows using more clusters if needed, each becoming more and more local in the data space.

In a finite mixture, the cluster indicator is drawn from a multinomial distribution with a Dirichlet prior, both having K possible values. At the intuitive level, the infinite case simply means allowing K to approach infinity. Since we only have N samples, there can in practice never be more than N clusters that would have observations. This makes the approach computationally possible, despite having an infinite number of parameters. More concretely, the infinite case is obtained via a Dirichlet process.

A Dirichlet process (DP) is a stochastic process for which the following property holds: For all probability distributions $G \sim \mathrm{DP}(G_0, \alpha)$, any partition $\{S_i\}_{i=1}^n$ of the event space satisfies

$$(G(S_1), ..., G(S_n)) \sim \mathrm{Dir}\left(\alpha G_0(S_1), ..., \alpha G_0(S_n)\right).$$

Here G_0 is the *base distribution*, $\alpha > 0$ is a *concentration parameter*, and $\mathrm{Dir}(\cdot)$ denotes the Dirichlet distribution. The DP was introduced in (Ferguson, 1973), and a more recent and machine-learning oriented presentation can be found for example in (Blei and Jordan, 2006). In the modern machine learning literature, DP is often interpreted through either the so-called Chinese Restaurant process (CRP) or the stick-breaking process. These are explicit presentations for processes that provide samples from the distribution G sampled from a DP, making the process more intuitively understandable. Here, the CRP interpretation is used to briefly describe the idea of DP in infinite mixture modeling.

Consider a case where samples $\boldsymbol{\theta}_i$ are drawn from the distributions G following the DP,

$$G \sim \mathrm{DP}(G_0, \alpha),$$
$$\boldsymbol{\theta}_i \sim G.$$

By integrating G out we reach the CRP description giving sampling formulas directly for $\boldsymbol{\theta}_i$ as

$$\boldsymbol{\theta}_1 \sim G_0,$$
$$\boldsymbol{\theta}_i|\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_{i-1} \sim \frac{\alpha}{i-1+\alpha} G_0 + \frac{1}{i-1+\alpha} \sum_{j=1}^{i-1} \delta_{\boldsymbol{\theta}_j},$$

where $\delta_{\boldsymbol{\theta}_j}$ is a delta-distribution centered on $\boldsymbol{\theta}_j$. These formulas illustrate that samples from CRP (and hence DP) are discrete, even if the base distribution was defined over an uncountable set (real space in the context of this thesis); a value sampled previously from the distribution has always finite probability of being sampled again. It also follows that the more often a sample has been chosen previously, the more likely it is to be sampled again. This is referred to as the *clustering property*. Note that despite the sampling scheme above being formulated as a sequential process, the order of sampling does not matter since samples from DP are infinitely exchangeable.

DP can be used to create infinite mixture models by a slight modification of the finite mixture model structure. In a finite mixture model, we have a single parameter vector $\boldsymbol{\theta}$ for each cluster in the model. However, in an infinite mixture model we directly draw a parameter vector for each data sample separately, using the CRP formulation. From the clustering property of DP, it follows that observations will still share their parameters in practice, and we can re-interpret each unique parameter value as one cluster in a mixture. This allows us to sidestep selecting the number of clusters, and moves the choice to the selection of G_0 and α. In practice, G_0 is simply the prior distribution for the cluster parameters, whereas α controls the clustering effect.

Implementing a Gibbs sampler for a finite mixture of CCAs is straightforward, though it is worth mentioning that the convergence of the naive implementation is often bad. One only needs to sample the cluster assignments for the data points, and then condition the sampling of the other parameters to the data points that are assigned to the particular cluster. Each cluster can here be treated independently, and hence having a mixture does not make this phase any more complex. The infinite mixture case is more difficult due to the constantly changing number of clusters, but still a relatively straightforward sampler is possible; we still have only a finite set of clusters at any given stage, and the only difference is that when changing the cluster assignments, we occasionally need to create or delete clusters. The exact details can be found for example in (Neal, 1998).

In practice, the naive sampler is inefficient both in the finite and infinite case, since sampling the cluster memberships one sample at a time makes it very difficult to, for example, move several samples together from one cluster to another. Practical sampling methods have been proposed to overcome this difficulty, but most of them only apply to models that have a complete conjugate prior (Jain and Neal, 2004). Here, however, the prior is conjugate only conditionally (i.e. the conditional distribution of a single parameter is conjugate, if all other parameters

are held fixed), which restricts the number of existing solutions. In Publication 5, a solution by Jain and Neal (2007) was chosen. The sampler alternates between normal Gibbs updates and updates where either splitting a cluster into two clusters or merging two existing clusters into a single one are proposed. A split/merge proposal is initiated by picking two samples, i and j. If they belong to the same cluster then a split is proposed, and otherwise merging of the two clusters is proposed. The idea in both operations is to use a restricted Gibbs sampling to obtain a good proposal; here the case of splitting is briefly explained. Two new clusters are sampled from the prior, and all of the samples in the cluster being splitted are assigned to either of these. Then an ordinary Gibbs sampler for finite mixtures is run only for the set of samples in the two clusters in question, in order to reach a good proposal that is more likely to be accepted. Finally, the proposal is either accepted or rejected based on the Metropolis-Hastings ratio (see, e.g., Gelman et al. (2003)). The actual sampling formulas can again be found in Publication 5.

It is worth mentioning that the ARD prior chosen for the Bayesian CCA (5.2) is particularly useful in the case of a mixture model. When optimizing a single Bayesian CCA, it would be possible to test the model with several different complexities (i.e. dimensionalities of the shared latent space), choosing the correct complexity based on the estimated marginal likelihood. For such a setup also other priors would be applicable. In the case of a mixture model, varying the complexity externally is not feasible, since even for a finite mixture of K clusters the number of potential latent space dimensionalities would be D^K, where D is the minimum of the data dimensionalities. The ARD prior allows using the full dimensionality for all clusters, while still getting results that only use a lower-dimensional subspace of the latent source when it is sufficient. It may be difficult to identify the exact number of components used for each cluster based on the posterior samples, but the model complexity will still be limited as if we had used lower-dimensional subspaces.

5.5 Generative dependency-seeking clustering models

5.5.1 Mixture of Gaussians with a shared latent source

Both projection and clustering methods maximizing the dependency were presented in Chapter 3. The probabilistic interpretation of CCA is an example of how to convert a projection method into a probabilistic alternative, and naturally the same can also be done for clustering models. In this section, a prototype clustering model stemming from the same idea is presented, originally proposed in Publication 3 and extended to a fully Bayesian case in Publication 4. It demonstrates how the probabilistic interpretation can be used to create new methods by a relatively simple change of distributional assumptions.

The probabilistic CCA was formulated as a model generating Gaussian noise on top of a linear transformation of a shared continuous latent variable. The model can be changed into a clustering model by changing the assumption for the latent shared space. In clustering models, each data point is assigned to one of K clusters, and thus the latent space z should consist of K possible discrete values. For maximal consistency, we could here have equal probabilities for the clusters, but in practice it makes more sense to allow clusters to have varying size. Hence,

the parameters of the distribution for z are free parameters.

The rest of the distributional assumptions should be chosen to match the assumptions on the data in question. We retain the assumption of the noise being Gaussian, largely for computational reasons, and also in order to have a connection to the widely used mixture of Gaussians (MoG) model (see McLachlan and Peel (2000)). The generative process is thus formulated as generating Gaussian noise around a cluster centroid specified by z. A convenient notational choice is to transform z into a K-dimensional binary vector $\tilde{\mathbf{z}}$ that has 1 as the zth element and zero as other elements. This allows directly sampling $\tilde{\mathbf{z}}$ from the multinomial distribution, as well as writing the centroid as $\mathbf{W}\tilde{\mathbf{z}}$. The discrete z can be regarded as picking a single column from the matrix $\mathbf{W}$.

The resulting model, after the marginalization of $\mathbf{z}_x$ and $\mathbf{z}_y$, is

$$\begin{aligned} \tilde{\mathbf{z}} &\sim \mathrm{Mult}(1, \boldsymbol{\alpha}), \\ \mathbf{x}|\tilde{\mathbf{z}} &\sim \mathrm{N}(\mathbf{W}_x\tilde{\mathbf{z}}, \boldsymbol{\Psi}_x), \\ \mathbf{y}|\tilde{\mathbf{z}} &\sim \mathrm{N}(\mathbf{W}_y\tilde{\mathbf{z}}, \boldsymbol{\Psi}_y). \end{aligned} \tag{5.3}$$

Alternatively, we can coerce the $\mathbf{x}$ and $\mathbf{y}$ into a single vector $\mathbf{v}$ by concatenation, leading to the model

$$\begin{aligned} \tilde{\mathbf{z}} &\sim \mathrm{Mult}(1, \boldsymbol{\alpha}), \\ \mathbf{v}|\tilde{\mathbf{z}} &\sim \mathrm{N}(\mathbf{W}\tilde{\mathbf{z}}, \boldsymbol{\Psi}), \end{aligned}$$

where

$$\boldsymbol{\Psi} = \begin{pmatrix} \boldsymbol{\Psi}_x & 0 \\ 0 & \boldsymbol{\Psi}_y \end{pmatrix}.$$

This is a classical MoG, with a special restriction on the covariance matrix of the clusters. It is assumed that the between-data covariance is zero within each cluster, whereas the within-data covariances are fully parameterized. Here, the covariance matrix $\boldsymbol{\Psi}$ is assumed to be the same for each cluster. By allowing $\mathbf{z}_x$ and $\mathbf{z}_y$, as well as $\mathbf{B}_x$ and $\mathbf{B}_y$, to depend on z (or equivalently $\tilde{\mathbf{z}}$) we can, however, extend the model to one with separate covariance matrices for the clusters. For a clustering model, this does not make the marginalization task (5.1) more difficult.

In Publication 3, the model (5.3) is optimized by maximizing the likelihood with an EM algorithm, and in Publication 4 the same model is extended to a Bayesian variant by introducing a variational Bayes approximation. The reader is referred to these publications regarding the details of the algorithms; the close connection to MoG allows re-using many of the formulas for the corresponding algorithms used for learning an ordinary MoG. Instead of VB, we could have used Gibbs sampling as in the case of the local dependent components, and in fact the sampling formulas of local dependent components can also be used to solve the clustering case with minor modifications (fix the projection matrix $\mathbf{W}$ of each cluster to zero, since no dependencies are allowed within the clusters). The formulation for the clustering case is here for a finite mixture, but the infinite case would be possible also in the variational approximation (Blei and Jordan, 2006; Kurihara et al., 2007).

The VB approximation is formulated as finding a factorized approximation $q(\boldsymbol{\theta}) = \prod_i q_i(\boldsymbol{\theta})$ for the true posterior $p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{Y})$. Here, $\boldsymbol{\theta}$ includes both the model parameters and the latent variables, and $q_i(\boldsymbol{\theta})$ are terms that each depend only on a part of the parameters; see Publication 4 for their form in this particular case. The approximation is fitted by minimizing the Kullback-Leibler divergence

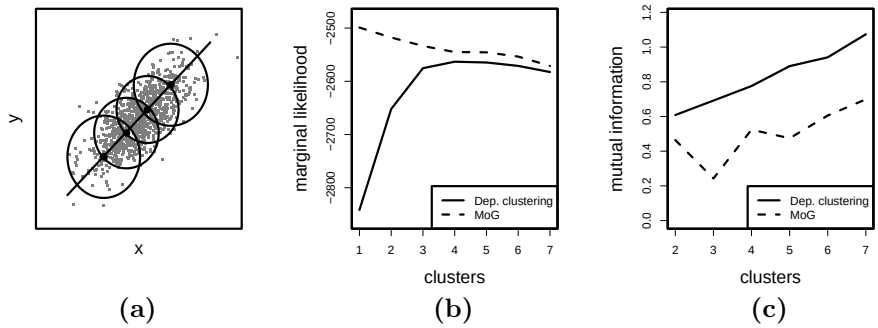

Figure 5.3: An illustration of how even data coming from a single Gaussian distribution needs to be modeled with a mixture of several Gaussians if attempting to detect dependencies with the cluster structure. **(a):** A scatterplot of two one-dimensional data sets, $\mathbf{X}$ and $\mathbf{Y}$, with clear dependency. The line depicts the CCA solution, and the ellipses are equiprobable contours of the Gaussians used as component densities. Note how the centroids of the clusters are perfectly aligned according to the linear dependency. **(b):** Comparison of lower bounds for the marginal likelihood given by an ordinary MoG (dashed line) and the method searching for dependencies (solid line); the ordinary MoG is clearly superior in representing the data for all numbers of clusters, the solution with a single cluster being the best. For the dependency-seeking model having more clusters is preferable, with 4 giving the optimal solution illustrated in **(a)**. **(c):** Comparison of the two methods in capturing the dependency with the cluster structure, showing that the proposed method (solid line) captures dependencies better than ordinary MoG (dashed line). The dependency is here measured with a contingency table-based estimate of mutual information between the cluster index and a discretized version of the mean of the canonical scores of $\mathbf{x}$ and $\mathbf{y}$. More of the dependency can be captured with a higher number of clusters. Notice that $K = 1$ is omitted, since the dependency measure is constant when all data is in a single cluster.

$d_{KL}(q, p)$, made computationally feasible by the factorization assumption. An alternative viewpoint is that VB maximizes a lower bound for the true marginal likelihood $p(\mathbf{X}, \mathbf{Y})$. The bound can be used to characterize how well the model describes the observed data.

The bound for the marginal likelihood is useful in demonstrating the compromise between finding the dependencies in the latent structure and describing the data as well as possible. In the following, a toy experiment is used to show how a traditional MoG can provide a better description for data, while still capturing the dependencies worse than the clustering model of (5.3). Consider two-dimensional multivariate normal data having high correlation between the two dimensions, and treat the dimensions as data sets $\mathbf{X}$ and $\mathbf{Y}$. Applying an unrestricted MoG to such data yields the natural result that a single cluster mixture model is the optimal solution. The dependency-seeking clustering model, however, finds more clusters since it needs to capture the correlation with z instead of the covariance matrix. As shown in Figure 5.3, the lower bound for the marginal likelihood is better for the ordinary MoG, even with cluster numbers differing from the optimal, and hence it is a better generative description for the data. However, only the model (5.3) captures the dependencies with the cluster indicator, as expected.

5.5.2 Structured dependencies

A fundamental restriction in the model structure of Figure 5.1 is that the dependencies are correctly captured in z only if the marginal models $p(\mathbf{x}, \mathbf{z}_x|z)$ and

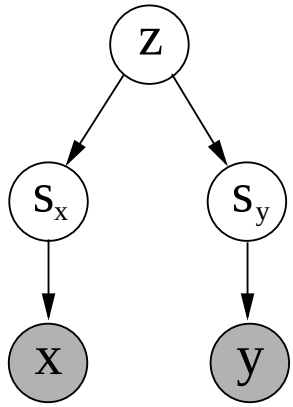

Figure 5.4: A graphical representation of the latent variable structure used in the clustering model that allows multimodal structure within the dependent clusters. Here, z is a latent variable determining the clusters, whereas s_x and s_y are used to model the multimodal structure within the clusters.

$p(\mathbf{y}, \mathbf{z}_y|z)$ are accurate, and if they are marginalized over $\mathbf{z}_x$ and $\mathbf{z}_y$. Satisfying this requirement is difficult when modeling data sets with complex within-data structure. A general-purpose model, such as a mixture of Gaussians, could in principle be used in place of the data set–specific models, but approximative marginalization techniques would be needed. As explained in Section 5.2.2, approximative marginalization of the data set–specific latent variables is left for future research.

In this section, a proof-of-concept implementation of an alternative solution to the same problem is presented. Instead of trying to build correct marginal models for complex data, the idea is to have more structure in the dependent part (Figure 5.4). In practice, the marginal models are assumed to be Gaussian to enable analytical marginalization, while a hierarchy of latent variables is built in place of z. The same model structure is presented also in (Bach and Jordan, 2005) as an alternative formulation for probabilistic CCA, but for Gaussian latent sources it does not make the model more flexible; instead, it is merely an equivalent formulation. In a clustering model, however, the hierarchy increases the capability of the model.

A two-level hierarchy of clustering-type latent variables is achieved by specifying the model as

$$\begin{aligned}
\tilde{\mathbf{z}} &\sim \mathrm{Mult}(1, \boldsymbol{\alpha}_z), \\
\tilde{\mathbf{s}}_x|\tilde{\mathbf{z}} &\sim \mathrm{Mult}(1, \boldsymbol{\Theta}_{s_x}\tilde{\mathbf{z}}), \\
\tilde{\mathbf{s}}_y|\tilde{\mathbf{z}} &\sim \mathrm{Mult}(1, \boldsymbol{\Theta}_{s_y}\tilde{\mathbf{z}}), \\
\mathbf{x}|\tilde{\mathbf{s}}_x &\sim \mathrm{N}(\mathbf{W}_x\tilde{\mathbf{s}}_x, \boldsymbol{\Psi}_x), \\
\mathbf{y}|\tilde{\mathbf{s}}_y &\sim \mathrm{N}(\mathbf{W}_y\tilde{\mathbf{s}}_y, \boldsymbol{\Psi}_y),
\end{aligned}$$

where $\tilde{\mathbf{s}}_x$ again denotes the binary encoding of s_x as a one-out-of-K vector. The variable to be interpreted as the final clustering solution is z, since it is the shared source. The other latent variables, s_x and s_y, merely act as nuisance parameters allowing the clusters to have multimodal structure in either or both spaces.

There is a relatively close connection between this model and the associative clustering method explained in Chapter 3. AC aims at maximizing the dependency

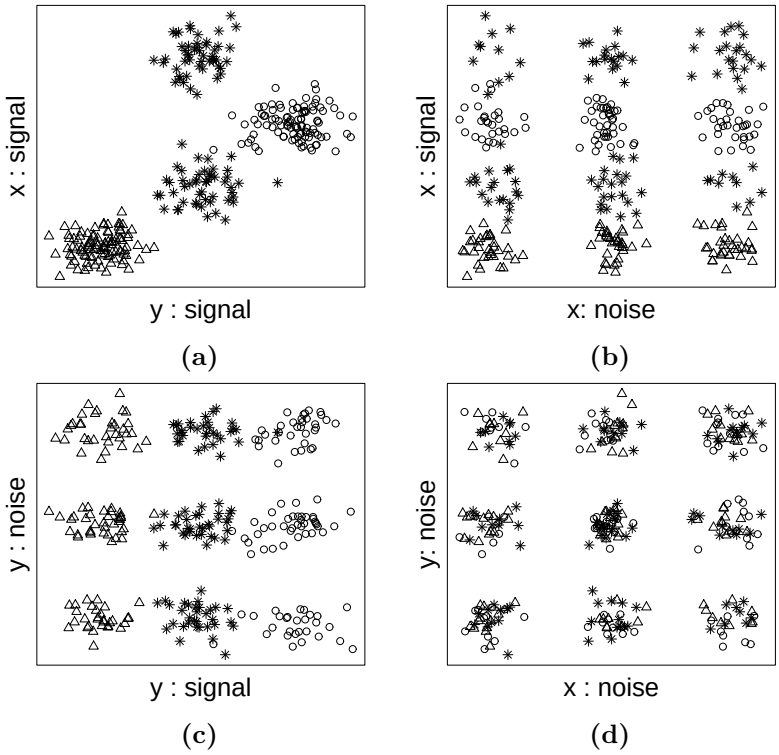

Figure 5.5: An illustration of dependency-seeking clustering of data with multimodal variation inside the clusters. Both **x** and **y** are two-dimensional, one dimension being dependent with the other data ("signal") and the other independent ("noise"). A three-cluster solution was sought, and the samples in all subfigures are marked with different symbols according to the cluster membership. **(a):** A scatterplot of the signal dimensions, showing how one of the clusters has multimodal structure even in the signal space. It is not divided into two, since the two data sets are independent within it. **(b)** and **(c):** Both **x** and **y** have multimodal structure in the noise direction, which is ignored by the method. **(d):** Looking at the scatterplot of the noise dimensions would suggest a high number of clusters, yet the actual relevant cluster structure is here not visible at all.

between the counts of samples in a contingency table, which could be interpreted probabilistically as maximizing the dependency between s_x and s_y in the model above. Here, the dependency is not explicitly maximized, but the common prior z restricts the form of $p(s_x, s_y)$. If searching for a small number of clusters, while allowing large number of values for s_x and s_y, we will necessarily get highly dependent s_x and s_y. Empirical comparison of these two methods is left for future research.

The model is illustrated in Publication 3, with simple two-dimensional data. The data has clear cluster structure, but the clusters have multimodal data set–specific structure. The illustration is reproduced in Figure 5.5, in a slightly extended form with more scatterplots. The method works well on this kind of toy data, but further development would be needed to make it a practical tool for data analysis. In particular, the EM algorithm used for optimization is very prone to local optima in this case. However, the model still works as a proof-of-concept that a probabilistic generative model can find the relevant cluster structure also

in the presence of complex marginal distributions.

5.5.3 Multi-view clustering

Multi-view clustering (Bickel and Scheffer, 2004, 2005) is a related clustering approach. Based on two co-occurring data sets, the goal is to cluster the samples so that the agreement between the sources is high. Using our notation, the proposed model assumes that $\mathbf{x}$ and $\mathbf{y}$ are independent given the cluster index z, just like (5.3). The proposed modeling approach is a kind of a hybrid between the approaches presented in Chapter 3 and here: The assumed model is generative and has a shared latent source, yet the learning is not based on optimization of the joint likelihood.

Denote by $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_y$ the parameters of the generative models for $\mathbf{x}$ and $\mathbf{y}$, respectively. The model is optimized by using the multi-view EM algorithm (also called Co-EM), which closely resembles the traditional EM algorithm. Formally, the algorithm maximizes a cost function that is a sum of standard likelihood terms for the separate views minus an additional penalty term that measures the disagreement. More intuitively, it can be thought of as the following alternating algorithm. The expectation step is done independently for the two spaces, computing $E[z|\mathbf{x}]$ and $E[z|\mathbf{y}]$ as separate steps. In the maximization step, however, the parameters $\boldsymbol{\theta}_x$ are optimized by maximizing the likelihood given the expected values $E[z|\mathbf{y}]$, and the parameters $\boldsymbol{\theta}_y$ given the expected values $E[z|\mathbf{x}]$. That is, the parameters are optimized using the expectation of the latent variables based on the other view. As a final result, a hard clustering is obtained by finding the most probable clustering based on both views.

Even though the model is formulated as a generative one, the optimization algorithm is somewhat heuristic. The algorithm is intuitively appealing, but not guaranteed to converge even to a local optimum like the traditional EM (though a converging variant can be constructed by annealing the disagreement term in the cost towards zero). In practical applications the algorithm seems to work well, even in situations where a single source is split randomly into $\mathbf{x}$ and $\mathbf{y}$, but for example an agglomerative clustering variant based on a similar idea does not have satisfactory performance (Bickel and Scheffer, 2004).

5.5.4 Dependency-seeking clustering of discrete data

The above clustering models work for continuous data, where data set–specific variation within each cluster follows the normal distribution. Dhillon et al. (2003) present a similar method for searching dependent clusters of discrete data by likelihood maximization. They consider the task of co-clustering, i.e., clustering both the rows and the columns of a joint probability distribution defined over two variables. In practice, a normalized contingency table is used as an estimate of the probability. Formulating the dependency clustering task through the co-clustering of the joint distribution is only possible in discrete domains, where enumerating all possible co-occurrence relations is possible. Nevertheless, the task is fundamentally the same as with the continuous clustering models: Cluster samples together so that the dependency between the cluster indices of the $\mathbf{x}$- and $\mathbf{y}$-spaces is maximized.

Since clustering can only decrease the mutual information, the task of finding maximally dependent clusters can alternatively be formulated through minimizing

the difference

$$I(X, Y) - I(S_x, S_y),$$

where S_x and S_y denote the random variables indicating the cluster memberships. Dhillon et al. (2003) show that in the case of hard clustering this is equivalent to assuming a probabilistic model of the form $p(x, y) = p(s_x, s_y)p(x|s_x)p(y|s_y)$ and finding the maximum likelihood solution of that, given that $p(x|s_x)$ and $p(y|s_y)$ are assumed to be correct. This follows the idea presented in Section 5.2; the density is factorized so that x and y only interact through the latent variables, and the noise process from the latent variables to the observations is assumed to be correctly modeled.

Chapter 6

Supervising unsupervised data analysis

6.1 Focusing analysis through dependencies

An interesting special case of the dependency maximization approach described in Chapter 3 is obtained by restricting the mapping for one of the data sets, here consistently $\mathbf{Y}$, to be fixed. This changes the task from a symmetric problem into a directed one, where the goal is finding a description for $\mathbf{X}$. The task is then comparable to traditional unsupervised learning, where we aim at understanding the structure in $\mathbf{X}$, but the learning is performed in such a manner that the dependencies between $\mathbf{X}$ and $\mathbf{Y}$ are emphasized. In other words, we only learn such structures of $\mathbf{X}$ that are informative of $\mathbf{Y}$. This focuses the analysis, and, in a sense, provides supervision in an unsupervised learning task.

The idea has a close connection to conditional modeling. When the mapping for the $\mathbf{y}$-space is fixed, the mutual information

$$I(S_x, S_y) = \int\int p(\mathbf{s}_x, \mathbf{s}_y) \log \frac{p(\mathbf{s}_x, \mathbf{s}_y)}{p(\mathbf{s}_x)p(\mathbf{s}_y)} d\mathbf{s}_x d\mathbf{s}_y$$

between the outputs S_x and S_y of the mappings reduces to the conditional entropy as

$$I(S_x, S_y) = \int p(\mathbf{s}_x, \mathbf{s}_y) \log p(\mathbf{s}_y|\mathbf{s}_x) d\mathbf{s}_x d\mathbf{s}_y - \int p(\mathbf{s}_y) \log p(\mathbf{s}_y) d\mathbf{s}_y.$$

Here, the latter term is simply the entropy of $S_y = Y$, which is a constant and hence does not affect modeling. For finite data, the first term reduces to the conditional log-likelihood

$$\log L(\boldsymbol{\theta}|\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{N} \log p(\mathbf{y}_i|\mathbf{s}_x^i, \boldsymbol{\theta}),$$

where $\mathbf{s}_x^i$ denotes a latent variable for $\mathbf{x}_i$, and $\boldsymbol{\theta}$ are the model parameters. Consequently, moving from the symmetric case to the directed case readily gives us a traditional probabilistic interpretation; instead of formulating the task as the maximization of mutual information between the latent variables, we can use the

conditional likelihood as the cost function. This simplifies the problem, and allows, for example, a direct application of Bayesian analysis.

A close connection to traditional supervised learning tasks, such as regression and classification (where $\mathbf{y}$ is a single categorical variable), is also worth noting. Despite sharing the same setting (paired $\mathbf{x}$ and $\mathbf{y}$, modeling based on the conditional distribution $p(\mathbf{y}|\mathbf{x})$), in dependency modeling the actual task is fundamentally different. In supervised analysis, the task is to predict $\mathbf{y}$, whereas here the task is to describe $\mathbf{x}$. In a sense, we are solving the unsupervised learning task in a setting borrowed from supervised analysis. In principle, it is also possible to borrow methods from the supervised learning, since some classification and regression models give a description of $\mathbf{x}$ as a side-product of the predictor. However, the accuracy of the description is usually not explicitly optimized, and hence methods directly targeted for analyzing $\mathbf{x}$ are preferred.

6.2 The learning metrics principle

The learning metrics (LM) principle (Kaski et al., 2001; Kaski and Sinkkonen, 2004; Sinkkonen and Kaski, 2002) explains how the dependencies between two data sets can be used to focus analysis of one of them, called the *primary data* and here denoted consistently by $\mathbf{x}$. The other data set, $\mathbf{y}$, is called the *auxiliary data*, and the task is to model the primary data in such a manner that it is informative of the auxiliary data. In the basic formulation, $\mathbf{y}$ is assumed to be a one-dimensional categorical variable (and is hence denoted by y from now on), but in principle this is not a necessary assumption.

In this setting, the formulation used in the rest of the thesis would thus be to find the latent variables $\mathbf{s}_x$, defined as mappings of $\mathbf{x}$, so that they have high dependency with y. The learning metrics principle provides an alternative formulation: Instead of constructing an explicit representation $\mathbf{s}_x$, we can change the metric of the data space so that conventional unsupervised learning methods will extract such mappings. In unsupervised learning, the choice of the metric is crucial but arbitrary; the results of many standard unsupervised learning methods are almost completely determined by the choice of the metric. By choosing the metric suitably, as shown by the LM principle, we can convert any unsupervised learning method into a one searching for dependencies.

Traditionally, different kinds of feature selection and weighting schemes are used to preprocess the data into such a form that using a simple metric, such as the Euclidean distance between the feature vectors, gives relevant results. For example, features known to be uninteresting are ignored, and the values of relevant features might be scaled up to emphasize them. The LM principle proposes an alternative to such manual preprocessing, defining the distance function directly in terms of the original features and using the auxiliary data to define the relevance. The metric gives small distances for differences in $\mathbf{x}$ on regions where the distribution of y remains constant, and large distances to variations on regions where the distribution of y changes considerably. That is, it reflects the changes in y.

Technically, the distance measure is based on the Kullback-Leibler divergence (3.1.2) between conditional distributions. Kullback-Leibler divergence as such is not a distance measure, since it is not symmetric and even the symmetrified variant does not satisfy the triangle inequality. It is, however, possible to define a real distance measure based on local divergences. Locally, for a small change $\mathbf{dx}$, the

squared distance is given as

$$d^2_{LM}(\mathbf{x}, \mathbf{x}+\mathbf{dx}) = d_{KL}(p(y|\mathbf{x}), p(y|\mathbf{x}+\mathbf{dx})),$$

where the Kullback-Leibler divergence is over two discrete distributions. The KL-divergence is zero if the two conditional distributions are identical, and if $p(y|\mathbf{x}+\mathbf{dx})$ differs greatly from $p(y|\mathbf{x})$ then the divergence is large. The metric can equivalently be formalized as defining the local metric matrix $\mathbf{J}(\mathbf{x})$, which allows expressing the local distance in the form

$$d^2_{LM}(\mathbf{x}, \mathbf{x}+\mathbf{dx}) = \mathbf{dx}^T \mathbf{J}(\mathbf{x}) \mathbf{dx}. \tag{6.1}$$

The metric matrix stemming from the Kullback-Leibler divergence is the *Fisher information matrix*, given as

$$\mathbf{J}(\mathbf{x}) = E_{p(y|\mathbf{x})} \left[\left(\frac{\partial}{\partial \mathbf{x}} \log p(y|\mathbf{x}) \right) \left(\frac{\partial}{\partial \mathbf{x}} \log p(y|\mathbf{x}) \right)^T \right].$$

Defining the metric locally is a crucial step. A global metric would correspond to just a linear preprocessing of features, since

$$d^2(\mathbf{W}\mathbf{x}_1, \mathbf{W}\mathbf{x}_2) = (\mathbf{W}\mathbf{x}_1 - \mathbf{W}\mathbf{x}_2)^T(\mathbf{W}\mathbf{x}_1 - \mathbf{W}\mathbf{x}_2) = (\mathbf{x}_1 - \mathbf{x}_2)^T \mathbf{W}^T \mathbf{W}(\mathbf{x}_1 - \mathbf{x}_2),$$

where $\mathbf{W}^T\mathbf{W}$ is the metric matrix. The locally defined metric, however, allows changing the structure of the data space by varying the importance of different features depending on $\mathbf{x}$. Most data analysis methods still require global distances, which are obtained by integrating the local distances from $\mathbf{x}_1$ to $\mathbf{x}_2$, traveling through the path that gives the smallest distance. In practice, this naturally requires approximations, such as the ones presented in Publication 6.

The learning metrics principle as such only characterizes the modeling task and defines the distance measure. The idea can actually still be implemented in various ways. In this thesis, two fundamentally different approaches are described. First, a kind of a plug-in method based on explicitly constructing an approximation of the global distances is presented, which allows using the new metric with most existing distance-based unsupervised learning methods. In the section thereafter, an approach that instead tailors a cost function to match the task of learning metrics is described, leading to a more computationally efficient method at the expense of generality.

6.2.1 Explicit metric estimation

The basic theory explained in the previous section, in principle, provides a metric that can be used as a distance measure in any learning algorithm that is based on distances between vectors in the $\mathbf{x}$-space. In other words, it provides an approach that can be used to change any unsupervised learning method into one that finds dependencies with y. However, the approach has two difficult phases that require a considerable amount of approximations and estimation.

First, the metric is defined through the Fisher information matrix that depends on the conditional distribution $p(y|\mathbf{x})$, which is naturally unknown. Yet we need to be able to evaluate it for any possible $\mathbf{x}$ considered by the algorithm. If we want to use the metric as a plug-in method for the distance calculation, then we need a method for estimating the conditional probabilities.

The second practical difficulty lies in computing the non-local distances. The global distance between vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ is given by the path integral of the local distance (6.1), over a path that gives the smallest distance. As we are working in an arbitrary real space with non-constant metric matrices, this is generally a very difficult problem.

In the remainder of this section, we first introduce a practical approximation developed for the explicit estimation of learning metrics distances, and then convert two standard unsupervised learning methods into dependency-seeking variants by using the learning metrics distance. Both the approximations and the resulting methods are introduced in Publication 6. Illustrations of the metric itself can be found for example in (Peltonen, 2004; Jensen, 2006). Jensen (2006) also compares the learning metrics principle to other approaches using adaptive local metrics, without using supervision from y. In a practical application of music retrieval, the learning metrics outperformed the unsupervised variants.

Approximations

To compute the metric matrix $\mathbf{J}(\mathbf{x})$ we need to have an estimate of $p(y|\mathbf{x})$ that can be evaluated for any $\mathbf{x}$. Furthermore, the estimate needs to be differentiable, since the expression of $\mathbf{J}(\mathbf{x})$ includes the gradient of the log-probability. The differentiability requirement rules out for example the density estimation methods based on binning.

A simple and intuitive choice is to construct a joint density estimate for $p(y, \mathbf{x})$, and to use the Bayes' rule (2.1) to derive $p(y|\mathbf{x})$. In Publication 6, estimates of the form

$$\hat{p}(y_i, \mathbf{x}) = \sum_{k=1}^{K} \pi_k \psi_{ki} e^{-\frac{\|\mathbf{x}-\boldsymbol{\mu}_k\|^2}{2\sigma^2}}$$

are used, where $p(y_i, \mathbf{x})$ indicates the probability of y paired with a particular $\mathbf{x}$ having value i, $\sum_i \psi_{ki} = 1$ parameterizes the distribution of y given component k, and $\sum_k \pi_k = 1$ are weights of the components. Furthermore, $\boldsymbol{\mu}_k$ are the cluster centroids for $\mathbf{x}$, and σ^2 is a shared variance parameter. In other words, we consider a generative model where the primary data comes from a mixture of (spherical) Gaussians, and each mixture component samples the auxiliary variables from a discrete distribution.

Such an estimate can be optimized either by maximizing the joint likelihood $\prod p(y, \mathbf{x})$, or by directly maximizing the conditional likelihood $\prod p(y|\mathbf{x})$. In the case of the joint likelihood, the model is called mixture discriminant analysis (MDA), and Hastie et al. (1995) present an EM algorithm for optimizing it. For directly optimizing the conditional likelihood, a gradient-based scheme optimization strategy is given in (Peltonen et al., 2002a). In Publication 6, it is shown that in practice using the conditional likelihood as the fitting criterion usually outperforms joint modeling, despite using a somewhat simplified model ($\pi_k = 1/K \quad \forall k$). In earlier work (Kaski et al., 2001; Peltonen et al., 2002a), also a Parzen-based estimate and an estimate formulated as a product of experts (Hinton, 1999) were used, but they are not considered here due to their computational complexity and poor performance.

Given a suitable density estimate, we can compute $\mathbf{J}(\mathbf{x})$ and thus define a local distance in a vicinity of any given $\mathbf{x}$. Unfortunately, this does not directly give global distances, and computing the actual path integrals is typically infeasible. It

is, however, relatively easy to devise approximations of varying complexity. Here, the examples considered in Publication 6 are summarized.

The simplest approach, introduced by Kaski et al. (2001), is to assume that when computing distances from a certain point $\mathbf{x}_1$, the metric matrix $\mathbf{J}(\mathbf{x})$ is constant and equals $\mathbf{J}(\mathbf{x}_1)$ everywhere. From this it follows directly that the shortest path is the straight line between the points, and the squared distance is given as the Mahalanobis distnance

$$d_1^2(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_2 - \mathbf{x}_1)^T \mathbf{J}(\mathbf{x}_1)(\mathbf{x}_2 - \mathbf{x}_1).$$

This approximation is only accurate for reasonably short distances or simple metrics.

A more accurate, but still computationally reasonably simple approximation is obtained by assuming that the shortest path equals the shortest path in the Euclidean metric (i.e. the straight line), but allowing $\mathbf{J}(\mathbf{x})$ to change along the path. This changes the integration into a one-dimensional integral, for which numerical integration is straightforward. In principle, any numerical integration method could be used; the simplest one is to split the line into T pieces and make the constant-value assumption within each piece. This gives

$$d_T(\mathbf{x}_1, \mathbf{x}_2) = \sum_{t=1}^{T} d_1(\mathbf{x}_1 + (t-1)/T\mathbf{v}_{12}, \mathbf{x}_1 + t/T\mathbf{v}_{12}),$$

where $\mathbf{v}_{12} = \mathbf{x}_2 - \mathbf{x}_1$, as the final distance. This is called the *T-point approximation*, noting that the local approximation is a special case of this with $T = 1$.

Finally, a yet more accurate approximation is obtained by also relaxing the assumption of a straight path. Given a set of points in the $\mathbf{x}$-space, we can compute all possible pairwise distances between the points (using the T-point approximation), and create a directed graph based on these. The points are nodes of the graph, and the distances define weights for the edges. The Floyd's algorithm, for example, can then be used to find the shortest path through such a graph. In principle, the set of nodes could be freely chosen, but in practice it is a good idea to use the actual data samples. This makes distance approximations more accurate in regions where the data is dense, and does not waste computational resources on areas with little data (where a density estimate would necessarily be inaccurate as well).

Self-organizing maps in learning metrics

Self-organizing map (SOM) (Kohonen, 2001) is an unsupervised neural network algorithm for visualizing and clustering vectorial data. The SOM consists of an ordered lattice, usually a two-dimensional grid, of units, also called model vectors. The data is represented by mapping each sample into the model vector that is closest to the sample, and the map is learned so that the model vectors close to each other on the lattice are similar. Even though the data representation is comparable to a simple clustering model, such as K-means, the SOM is a stronger data analysis tool as it aims to retain the topology of the data through the ordering on the lattice.

The simplest way to learn a SOM is to use an iterative learning procedure, where a single data point is presented to the map at a time. First, the closest

model vector, the best matching unit (BMU), is selected by

$$w(\mathbf{x}) = \arg\min_i d(\mathbf{x}, \mathbf{m}_i),$$

where $\mathbf{m}_i$ denotes the ith model vector, and $d(\cdot, \cdot)$ is some distance measure, typically the Euclidean distance. After selecting the BMU, the model vectors in the lattice are updated towards the data sample, so that the BMU is moved most. Also nodes that are close to the BMU on the lattice are moved, and their updates are governed through a *neighborhood function* $h(i, j)$. The update rule can be written as

$$\mathbf{m}_j(t+1) = \mathbf{m}_j(t) + \alpha(t)h(w(\mathbf{x}), j)(\mathbf{x} - \mathbf{m}_j(t)),$$

where α is a learning parameter which is slowly decreased during learning. The neighborhood function is a decreasing function of the distance in the lattice, for example a Gaussian. In practice, a more efficient algorithm, called the batch algorithm, is obtained by updating the model vectors only after mapping all of the data samples, but the basic idea remains the same.

Learning a SOM in the learning metrics is intuitively straightforward. As the algorithm is based on simply finding the closest model vector and then updating the model vectors, all we need to do is to replace the distance function in the BMU selection by the learning metrics distance. In addition, the update rule needs to be replaced by the natural gradient (Amari, 1998), but it turns out (Kaski et al., 2001) that the actual update rule is still identical for the approximations assuming a straight line for the shortest path.

A remaining step is to choose which approximations to use. In the iterative SOM algorithm, each iteration involves M distance computations, where M is the number of map units. Each of these is computed starting from the same point, $\mathbf{x}$. Hence, computing the one-point distance where the metric matrix is assumed to be constant is very efficient, as the metric matrix needs to be evaluated only once. This distance approximation was used in (Kaski et al., 2001), together with a density estimate based on maximizing the joint density. In Publication 6, it is shown that, in practice, better results are obtained by optimizing the density estimate using the conditional likelihood, and that using the more complex T-point distance approximation also provides more accurate results. The computational burden can be reduced, if necessary, with a speed-up where more accurate distances are only computed to the best W candidates chosen based on the simpler approximation.

The SOM in learning metrics can be applied to most scenarios where the classical SOM is applicable. Considering the huge literature of SOM applications (see Kaski et al. (1998); Oja et al. (2003); Pöllä et al. (2006)), it is not worthwhile to list those here, but it is good to make a remark about the practical restrictions. First, in order to be able to use the learning metrics principle, one needs to be able to give the additional information in the form of a categorical variable, typically as some kind of a class variable. The second restriction comes from computational cost; it is not feasible to estimate the T-point distances for large data collections or maps. In practice, SOM in learning metrics has been applied for the analysis of financial data (Kaski et al., 2001) and in bioinformatics applications (Kaski et al., 2003).

Multidimensional scaling in learning metrics

Metric multidimensional scaling (see Borg and Groenen (1997)) methods are used to visualize high-dimensional data sets in a low-dimensional space, usually on a two-dimensional plane. The idea is to find new representations for the samples so that pairwise distances are preserved as accurately as possible. Typically, the approach is non-parametric in the sense that no assumptions on the mapping are made. Instead, the locations of all representations are optimized independently.

Sammon's mapping (Sammon, Jr., 1969) is a typical MDS method, defined with the cost function

$$E_S = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \frac{(d_{ij} - o_{ij})^2}{d_{ij}}$$

where d_{ij} denotes the distance between samples i and j in the original data, and o_{ij} is the distance between the samples in the output space. As the distances are scaled with the original distances, the method focuses on getting the shortest distances modeled accurately. Given a matrix of pairwise distances in the original data space, the mapping can be computed by minimizing the cost E_S by, for example, the steepest-descent method or other gradient-based methods.

Moving from a standard, Euclidean, Sammon's mapping to Sammon's mapping in learning metrics is again conceptually easy. All we need to do is to replace the distances d_{ij} by those that are computed in learning metrics. As each distance needs to be computed only once, it is feasible to use more accurate approximations compared to the SOM case. In particular, using the graph-based approximation allowing piecewise linear shortest paths is now computationally possible. In Publication 6, Sammon's mapping in learning metrics is studied both with the T-point approximation and the graph-based approximation, using different values of T. The graph approximation provides consistently, though not by a very wide margin, better results in keeping the samples of the same class close to each other. Both approximations clearly outperform the Sammon's mapping in Euclidean metric, which is understandable as the standard mapping ignores the class information completely.

The Sammon's mapping is here illustrated from a point of view that aims at representing general properties of the learning metrics idea. In Figure 6.1, the learning metrics and Euclidean metric are compared in the task of finding a one-dimensional mapping of two-dimensional data. In addition, a comparison metric that uses the auxiliary information to define global distances directly based on the Kullback-Leibler divergence is presented in order to demonstrate the importance of defining the metric locally. For details of the comparison method, see Publication 6.

6.2.2 Dependencies through conditional density

Discriminative clustering

An alternative approach to learning metrics is to start from the same setting and to construct a method that achieves similar properties without explicitly approximating distances in the learning metrics. One such approach is taken in *discriminative clustering* (DC) (Sinkkonen and Kaski, 2002), which seeks to cluster the primary data so that the cluster indices are informative about the auxiliary variable. This approach resembles more closely the approach of Chapter 3, since the dependency

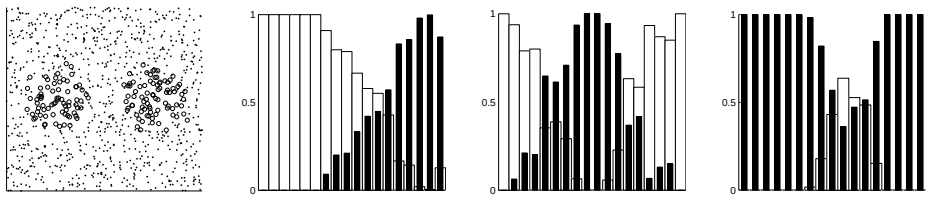

Figure 6.1: An illustration of learning metrics and of the importance of defining the metric locally. The subfigure **(a)** presents two-dimensional data set with two possible auxiliary variables marked with different symbols. The dots represent a background class, whereas the circles mark the other, here called foreground, class. The task is to visualize the data with a one-dimensional mapping so that the areas of homogeneous auxiliary variables are displayed together, which is satisfied with the learning metrics result in subfigure **(c)**; the two clusters of the foreground class are separated to opposite ends of the visualization, whereas the background class is grouped together in the middle. A comparison method using the class information but ignoring the topology, shown in **(b)**, groups the two separate regions of the foreground to one end and the background to the other end. It has thus lost the topological knowledge of the class being separated into two parts. The Sammon's mapping in Euclidean metric, shown in **(d)**, is not able to make any clear separation between the classes, as expected. The subfigures **(b-d)** present normalized histograms of the two classes along the one-dimensional Sammon's mapping. Black bars correspond to the background samples, whereas white bars are for the foreground samples.

between y and a mapping of $\mathbf{x}$ is explicitly measured and maximized. However, it is still possible to prove a connection to the learning metrics.

The clustering model is defined as the classical K-means clustering: Each sample is assigned to the cluster whose prototype vector $\mathbf{m}_i$ is the closest. Hence, each cluster prototype defines a Voronoi region V_i, i.e. a region in the data space where that prototype is closer than any other prototype. Each cluster also has an associated *distributional prototype* ψ_i, which is a discrete distribution for the auxiliary variable within that cluster. The model is thus a piecewise-constant generative model for y conditioned on $\mathbf{x}$.

The original formulation of the algorithm (Sinkkonen and Kaski, 2002) considers the task of minimizing the Kullback-Leibler divergence between the prototypes and the true conditional distribution, averaged over the Voronoi regions, that is

$$E_{SDC} = \sum_{i=1}^{K} \int_{V_i} d_{KL}\left(p(y|\mathbf{x}), \psi_i\right) p(\mathbf{x}) dx. \tag{6.2}$$

The cost is optimized with a stochastic gradient method, and SDC in the cost stands for stochastic DC. This is equivalent to maximizing the mutual information between the cluster indices and the auxiliary variable. It has also been shown (Kaski and Sinkkonen, 2004) that the task is asymptotically equivalent (for large data) to performing vector quantization in the learning metrics, with the restriction that the Voronoi regions are still defined in the Euclidean metric. It is possible to relax that restriction, but it leads to a rather computationally heavy algorithm (Salojärvi et al., 2003). Extending the model outside real spaces is also possible; Peltonen et al. (2002b) present a discriminative clustering model for text documents represented in the bag-of-words form.

In practice, we need not consider the formulation involving Kullback-Leibler divergence, but we can instead use traditional probabilistic learning. Maximiz-

ing the conditional likelihood of the implied generative model is asymptotically equivalent to (6.2), but the likelihood is more justified and easier to handle for finite data. Using the conditional likelihood as the optimization criterion, as is done in Publication 7, also makes the connection to the rest of the thesis more explicit, since the optimization criterion is directly the dependency between the cluster index and the auxiliary variable.

The final outcome of the model is the clustering of the primary data space, and thus the distributional prototypes ψ_i are not actually needed. Following the standard Bayesian approach, we integrate such nuisance parameters out, which can be done analytically by assuming a Dirichlet prior. The integration leads to the marginalized conditional log-likelihood

$$E_{DC} = \sum_{i,j} \log \Gamma(n_j^0 + n_{ij}) - \sum_j \log \Gamma(N^0 + N_i) + \text{constant}, \tag{6.3}$$

which can also be interpreted as the posterior distribution of the cluster centroids if an improper prior $p(\mathbf{m}) \propto 1$ is used. Here, n_{ij} denotes the number of samples in cluster i having the jth auxiliary value, and $N_i = \sum_j n_{ij}$. $N^0 = \sum_j n_j^0$ are the parameters of the Dirichlet prior. As shown in (Sinkkonen et al., 2002), this formulation is equivalent to maximizing a Bayes factor (3.3) measuring the dependency between the cluster indices and the auxiliary data.

The cost function only depends on the counts of samples within the clusters, which makes it non-differentiable and, consequently, difficult to optimize. This is overcome by introducing smoothed membership functions

$$y_i(\mathbf{x}) \propto \exp\left(-\frac{\|\mathbf{x} - \mathbf{m}_i\|^2}{2\sigma^2}\right)$$

that result in the final cost function being expressed in terms of smoothed counts $\tilde{n}_{ij}$, computed by summing $y_i(\mathbf{x})$, instead of the counts of the discrete occurrences. This allows computing the gradient with respect to the model vectors, and hence the optimization with any gradient-based optimization method. In Publication 7, it is shown that gradient-based optimization of the smoothed version gives comparable or better results than what is obtained by directly maximizing the non-differentiable cost with an approach based on simulated annealing (Kirkpatrick et al., 1983), but it is computationally much more efficient. Both algorithms operating on the marginalized cost (6.3) also outperformed the original stochastic algorithm of Sinkkonen and Kaski (2002) in most scenarios.

In Publication 7, we also present two regularization approaches to control the overfitting of the DC model. The first stems from a practical observation made during the development and use of the model; the algorithm occasionally leads to local optima where some prototype vectors are pushed away from the actual data, eventually resulting in them containing few or no data samples. This is an undesirable property if the target is to obtain a certain number of clusters, and thus a straightforward regularization can be implemented by penalizing such solutions. In practice, the penalization is done by adding a term measuring the entropy of the number of samples in clusters, weighted by a free parameter controlling the amount of regularization.

A theoretically more interesting regularization method is obtained by changing the modeling task so that is also takes the primary data into account. The basic DC cost (6.3) only involves the conditional distribution $p(y|\mathbf{x})$, ignoring the density

of $\mathbf{x}$ within the clusters. Changing from a conditional model to a generative model for both $\mathbf{x}$ and y,

$$p(y, \mathbf{x}|\{\mathbf{m}\}, \{\psi\}) = p(y|\mathbf{x}, \{\mathbf{m}\}, \{\psi\})p(\mathbf{x}|\{\mathbf{m}\}),$$

allows taking also the structure of $\mathbf{x}$ into account. In Publication 7, a mixture of Gaussians (MoG) is used as $p(\mathbf{x}|\{\mathbf{m}\})$. Gaussians with a shared covariance matrix of the form $\lambda\mathbf{I}$ are used, both to simplify the computation and to yield a model with only one tuning parameter, λ. A larger λ means that less weight is given for the regularizer, with $\lambda = \infty$ corresponding to the unregularized version.

Interestingly, the shift from the conditional model to the joint model in the directed case can be paralleled to changing from the explicit dependency maximization techniques (Chapter 3) to searching for symmetric dependencies using generative models (Chapter 5). In both cases, the former approach ignores a part of the variation in the data collection, the distribution of the data given the latent variable, while the latter models all observed data.

In Chapter 5, we explained that a joint model only finds dependencies correctly when the distributions of $\mathbf{x}$ and $\mathbf{y}$ given the latent variables are correct. In the case of DC, the equivalent of that requirement would be that the generative model $p(\mathbf{x}|\{\mathbf{m}\})$ needs to be correct, and otherwise moving from the pure DC cost to the joint model will eventually result in an incorrect solution if the amount of regularization is increased. With the MoG regularization, the assumption is that the data follows the normal distribution within each cluster, which is a too restricted assumption, especially when the covariance is assumed spherical and identical for all clusters. Furthermore, to move from a regularization interpretation to a model equivalent to the joint approach in the symmetric case, we would also need to learn the covariances of the mixture model to fit the data.

In fact, it turns out that the variational Bayes clustering model presented in Chapter 5 can be used to solve a task relatively similar to DC. If the auxiliary variable y is continuous instead of discrete, the clustering model is readily applicable. It clusters $\mathbf{x}$ and y jointly so that the covariance between $\mathbf{x}$ and y is forced to be zero, resulting in a clustering result that closely corresponds to the DC solution. It is still assumed that the data within each cluster comes from a normal distribution, but now the covariance matrix is fully parameterized. Modifying the method to allow discrete y would be relatively straightforward, essentially leading to a Bayesian version of the mixture discriminant analysis by Hastie et al. (1995). Interestingly, MDA (with restricted covariances) is used as a comparison method for DC in Publication 7, where it was shown to be inferior compared to DC in capturing the dependency between $\mathbf{x}$ and y with the cluster structure. Relaxing the restrictions on the covariances while also having a fully Bayesian method could, however, provide a practically applicable method.

Informative components

Also linear components aiming at capturing dependencies with an auxiliary variable have been presented. Some methods for the task are discussed here, due to the close connection to the CCA approach in the symmetric case.

The classical unsupervised linear projection method is principal component analysis (PCA) (Hotelling, 1933), which searches for projection directions with maximal variance. For supervised analysis the corresponding method is linear discriminant analysis (LDA) (Fisher, 1936), which finds linear projections maximally

separating classes assumed to follow the normal distribution. Between these two extremes lies the supervised unsupervised variant, here called *informative components* (Peltonen and Kaski, 2005). It searches for linear projections of $\mathbf{x}$ such that the dependency between the projections and the categorical y is maximized.

The dependency is here measured with the conditional likelihood, like in the case of DC. As y is fixed, it corresponds to maximizing the mutual information. In (Peltonen and Kaski, 2005) and (Goldberger et al., 2005), approaches based on non-parametric estimation were presented, using Parzen kernel estimates for predicting the class densities. Peltonen and Kaski (2005) also show how the method is connectd to the learning metrics principle. These methods are direct counterparts for the symmetric DeCA presented in Chapter 3 and Publication 1. Peltonen et al. (2007) introduce a version using a MoG for the density estimation, instead of Parzen kernels, to improve computation time for large data sets. Another related method, also based on Parzen kernels but using a different approximation for mutual information, is given in (Torkkola, 2003).

6.3 Other approaches

6.3.1 Information bottleneck

In Chapter 3, the information bottleneck (IB) framework was mentioned when discussing approaches to the symmetric case. IB was originally proposed for the directed case, and is hence treated here in slightly more detail.

The basic IB (Pereira et al., 1993; Tishby et al., 1999) considers the task of clustering a single variable X so that the cluster indices S_x are informative of Y. The fundamental task is thus the same as in DC. The main difference is that IB is defined for discrete variables, and can thus use a somewhat different formulation for the actual clustering criterion. In DC, it was assumed that samples falling into a single Voronoi region are clustered together, thus defining an explicit structure in the input space, whereas IB considers the task of clustering as the minimization of the mutual information between the cluster indices and the original variable.

In general, a clustering task formulated as minimizing $I(X, S_x)$ needs to be constrained in order to avoid the solution of having all data grouped into a single cluster. This can be done by adding a term that measures the distortion between X and S_x. The distortion measure can be chosen in several ways, and in IB the distortion is extracted from the auxiliary variable. The task is then to minimize the cost

$$E_{IB} = I(X, S_x) - \beta I(S_x, Y),$$

which is essentially a tunable compromise between compressing X and being informative of Y. A value of $\beta = 0$ reduces this to pure clustering task with no restrictions on the quality (and no connection to Y), whereas $\beta = \infty$ corresponds to maximizing the mutual information between the clusters and the auxiliary data. That is, with $\beta = \infty$ the task is the same as in DC. In DC the solution is constrained by the topology of the continuous $\mathbf{x}$-space, whereas IB requires finite β to constrain the solution through the first term in the cost E_{IB}.

The IB clustering can be solved with several methods, for example by using an agglomerative algorithm by Slonim and Tishby (2000), or a sequential algorithm by Slonim et al. (2002). Peltonen et al. (2004) extend the latter to the small sample case, by replacing the mutual information with the Bayes factor.

6.3.2 Clustering with pairwise constraints

The fundamental task discussed in this chapter is focusing unsupervised learning on more relevant ascpects of the data. In the learning metrics principle, the relevance is determined through dependencies between $\mathbf{x}$ and y, but it is also possible to use other forms of prior information. One alternative approach is based on explicitly constraining the solutions of the learning methods based on pairwise relations between the data samples, which has been extensively studied in the case of clustering models (Basu et al., 2002; Lu and Leen, 2007a,b). If we know that a certain relation between two samples holds, then a result that retains the relation is likely to be more informative.

Many clustering methods using pairwise relations use the same basic approach. The clustering of $\mathbf{x}$ is constrained by additional information indicating that certain samples need to be in the same cluster (must-link) or that they are not allowed to be in the same cluster (cannot-link). These constraints may be either hard (Basu et al., 2002) or soft (Lu and Leen, 2007b). Typically, the models are formulated as extensions of the K-means of MoG, but also methods based on discriminative learning and more complex latent variable structures have been proposed (Lu and Leen, 2007a).

Intuitively, these methods are relatively close to DC, both seeking to cluster $\mathbf{x}$ given some additional information, and the main difference lies in the form of the additional information. Fundamentally, the feasibility of the approaches depends on the nature of the application and on the available prior information. From a full classification, it would be possible to derive a set of soft constraints, allowing the use of constrained clustering algorithms to mimic DC, but it is unlikely that such a solution would be better in practice. DC, on the other hand, is not applicable to situations where only a potentially small number of pairwise constraints is available.

Chapter 7

Summary and conclusions

In this thesis, we consider the task of focusing exploratory data analysis methods to properties deemed relevant by the person doing the analysis. The relevance is here determined through statistical dependencies between data sources, based on the assumption that the information shared by two or more sources chosen by the analyst is more relevant. We cover two different data analysis settings in which this idea can be applied, discussing the theory behind these approaches and introducing several practical methods that stem from the underlying theory.

The more straightforward setting is: Analyze a single multivariate data source so that the dependencies between the samples and a categorical auxiliary variable are emphasized. That is, we assume that structure in the data being analyzed is more relevant if there is corresponding structure also in the auxiliary source. In this thesis, three different methods based on the learning metrics principle are introduced and empirically compared to alternative approaches. The learning metrics principle uses the dependencies to define a new distance measure, which can be used with most unsupervised learning methods. Two of the methods, discriminative clustering and self-organizing map in learning metrics, are not novel, but both are improved considerably in this work by providing better approximations and optimization schemes, yielding more accurate results. The third method, Sammon's mapping in learning metrics, is novel.

In the other setting, the task is to symmetrically analyze two or more data sources with co-occurring samples. The relevance is defined as the variation shared by the sources, and the dependencies are used to formulate a data fusion approach for exploratory data analysis: Combine the data sources so that variation in common between them is enhanced. An alternative view is that the data sets supervise each other. Two different learning strategies for this task are considered. First, the task is formulated as finding representations that maximize a dependency criterion. A novel method of dependent component analysis is proposed, and a way of using canonical correlation analysis as a preprocessing method in a data fusion setting is discussed. Second, Bayesian generative models that capture the dependencies are introduced. We first present a general model structure and discuss the conditions under which the models will find the dependencies, and then introduce three novel models for different settings. Bayesian generative models for detecting statistical dependencies in this sense have not been considered before.

As explained above, a number of novel and practically applicable methods and models are introduced. These methods are summarized in Table 7.1, listing for

Method	Setting	Criterion	Chap.	Publ.
Dependent component analysis	symmetric	non-parametric estimate of mutual information	3	1
CCA for data fusion	symmetric	generalized correlation	4	2
Probabilistic/Bayesian CCA	symmetric	likelihood/posterior	5	3, 4, 5
Dependency clustering	symmetric	likelihood/posterior	5	3 and 4
Local dependent components	symmetric	posterior	5	5
SOM in learning metrics	directed	local metric	6	6
Sammon's mapping in learning metrics	directed	local metric	6	6
Discriminative clustering	directed	conditional likelihood	6	7

Table 7.1: A summary of the novel computational methods and models presented in the thesis. For each method the setting (symmetric or directed) and the optimization criterion are listed, as well as the chapter and publication discussing the method. Strictly speaking, "local metric" is not an optimization criterion, but it refers here to the fact that the novel contribution in those methods is an application of a locally defined metric. The cost function remains the same as it is for the traditional versions of the corresponding methods. "Posterior" refers to the method being based on approximating the posterior distribution of the whole solution space. At a more detailed level, the Bayesian CCA and the local dependent components use Gibbs sampling for inference, while the clustering model uses variational approximation.

each method the setting and the optimization criterion, together with instructions on where to find the description of the method. The methods are listed in the order they are discussed in this introductory part of the thesis.

7.1 Future research directions

The work done in the thesis has two obvious main future directions. First, the theoretical work could be continued, and secondly, the methods could be applied into practical real-life data analysis cases. Some potential application fields were listed in Chapter 3, and all of the methods developed in the thesis would fit to many of those applications. Here, we concentrate on the potential advances on the theory side.

An obvious extension would be to continue the work on the Bayesian theory of dependency modeling. In this thesis, the data set–specific variation is always assumed simple enough, so that we can marginalize the data set–specific latent variables out in closed form. The possibility of using approximate marginalization, utilizing any of the available Bayesian inference techniques, would need to be studied. With approximative marginalization, considerably more complex model families could be considered.

At the level of practical methods, the thesis calls for further development in at least the following scenarios. First, the dependent component analysis method described in Section 3.2.2 could be improved by considering parametric estimates

instead of non-parametric ones, like was done in (Peltonen et al., 2007) for the non-symmetric variant of the method, allowing faster optimization and hence wider applicability. Work on this is currently in progress. Second, a fully Bayesian generative treatment of the discriminative clustering (Section 6.2.2) would provide a novel version of the mixture discriminant analysis. Finally, novel dependency-seeking generative models could be derived by changing the distributional assumptions, made possible by the approximative marginalization of data set–specific latent variables.

Bibliography

S. Akaho. A kernel method for canonical correlation analysis. In *Proceedings of the International Meeting of the Psychometric Society (IMPS2001)*. Springer-Verlag, 2001.

S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10:251–276, 1998.

C. Archambeau, N. Delannay, and M. Verleysen. Robust probabilistic projections. In W. Cohen and A. Moore, editors, *Proceedings of the 23rd International conference on machine learning*, pages 33–40. ACM, 2006.

F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

F. R. Bach and M. I. Jordan. A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Department of Statistics, University of California, Berkeley, 2005.

S. Basu, A. Bannerhee, and R. Mooney. Semi-supervised clustering by seeding. In C. Sammut and A. Hoffmann, editors, *Proceedings of the 19th International Conference on Machine Learning*, pages 19–26, 2002.

T. Bayes. Studies in the history of probability and statistics: IX. Thomas Bayes' essay Towards solving a problem in the doctrine of chances. *Biometrika*, 45:296–315, 1763/1958.

S. Becker. Mutual information maximization: models of cortical self-organization. *Network: Computation in Neural Systems*, 7:7–31, 1996.

S. Becker and G. E. Hinton. Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355:161–163, 1992.

J. Beirlant, E. Dudewicz, L. Györfi, and E. van der Meulen. Nonparametric entropy estimation: An overview. *International Journal of the Mathematical Statistics Sciences*, 6:17–39, 1997.

J. A. Berger, S. Hautaniemi, S. K. Mitra, and J. Astola. Jointly analyzing gene expression and copy number data in breast cancer using data reduction models. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(1):2–16, 2006.

J. M. Bernardo and A. F. M. Smith. *Bayesian Theory*. John Wiley & Sons Ltd, Chichester, England, 1994.

S. Bickel and T. Scheffer. Multi-view clustering. In *Proceedings of the IEEE International Conference on Data Mining*, 2004.

S. Bickel and T. Scheffer. Estimation of mixture models using Co-EM. In *Proceedings of the European Conference on Machine Learning*, 2005.

T. D. Bie and B. D. Moor. On the regularization of canonical correlation analysis. In S.-I. Amari, A. Cichocki, S. Makino, and N. Murata, editors, *Proceedings of the International Conference on Independent Component Analysis and Blind Source Separation (ICA2003)*. 2003.

C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

C. M. Bishop. Latent variable models. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 371–404. MIT Press, 1999a.

C. M. Bishop. Bayesian PCA. In M. S. Kearns, S. Solla, and D. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11, pages 382–388. MIT Press, 1999b.

D. M. Blei and M. I. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian analysis*, 1(1):121–144, 2006.

I. Borg and P. Groenen. *Modern Multidimensional Scaling*. Springer, New York, 1997.

M. Borga. Canonical correlation - a tutorial. http://people.imt.liu.se/∼magnus/cca/, 2001.

O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised learning*. MIT Press, 2006.

G. Chechik, A. Globerson, N. Tishby, and Y. Weiss. Information bottleneck for Gaussian variables. *Journal of Machine Learning Research*, 6:165–188, 2005.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.

C. Dehon, P. Filzmoser, and C. Croux. Robust methods for canonical correlation analysis. In *COMPSTAT: Proceedings in Computational Statistics*, pages 321–316, 2000.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977.

I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *Proceedings of KDD'03, The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 89–98. ACM Press, New York, NY, USA, 2003.

R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley & Sons, 1999.

B. Efron. Bayesians, frequentists, and scientists. *Journal of the American Statistical Association*, 100(469):1–5, 2005.

B. Efron. Bootstrap methods: another look at the jackknife. *Annals of Statistics*, 7:1–26, 1979.

J. D. R. Farquhar, D. R. Hardoon, H. Meng, J. Shawe-Taylor, and S. Szedmak. Two view learning: SVM-2K, theory and practice. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 355–362, Cambridge, MA, 2006. MIT Press.

T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1(2):209–230, 1973.

X. Fern, C. E. Brodley, and M. A. Friedl. Correlation clustering for learning mixtures of canonical correlation models. In H. Kargupta, C. Kamath, J. Srivastava, and A. Goodman, editors, *Proceedings of the fifth SIAM International conference on data mining*, pages 439–448, 2005.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

J. W. Fisher III and T. Darrell. Speaker association with signal-level audiovisual fusion. *IEEE Transactions on Multimedia*, 6(3):406–413, 2004.

N. Friedman, O. Mosenzon, N. Slonim, and N. Tishby. Multivariate information bottleneck. In *Proceedings of UAI'01, The Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 152–161. Morgan Kaufmann Publishers, San Francisco, CA, 2001.

O. Friman, J. Cedefamn, M. Borga, P. Lundberg, and H. Knutsson. Detection of neural activity in fMRI using canonical correlation analysis. *Magnetic Resonance in Medicine*, 45(2):323–330, 2001.

K. Fukumizu, F. Bach, and A. Gretton. Statistical consistency of kernel canonical correlation analysis. *Journal of Machine Learning Research*, 8:361–383, 2007.

C. Fyfe and P. Lai. ICA using kernel canonical correlation analysis. In *Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation (ICA 2000)*, pages 279–284, 2000.

C. Fyfe and G. Leen. Stochastic processes for canonical correlation analysis. In *Proceedings of 14th European Symposium on Artificial Neural Networks*, pages 245–250, 2006.

F. Galton. Regression towards mediocrity in hereditary stature. *Journal of the Anthropological Institute*, 15:246–263, 1886.

A. Gelfand and A. Smith. Sampling-based approaches to calculating marginal densities. *Journal of American Statistical Association*, 85:398–409, 1990.

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis (2nd edition)*. Chapman & Hall/CRC, Boca Raton, FL, 2003.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 513–520. MIT Press, Cambridge, MA, 2005.

I. J. Good. On the application of symmetric Dirichlet distributions and their mixtures to contingency tables. *Annals of Statistics*, 4(6):1159–1189, 1976.

A. Gretton, R. Herbrich, A. Smola, O. Bousquet, and B. Schölkopf. Kernel methods for measuring independence. *Journal of Machine Learning Research*, 6:2075–2129, 2005.

D. L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEE*, 85(1):6–23, 1997.

D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

D. R. Hardoon, J. Mourão-Miranda, M. Brammer, and J. Shawe-Taylor. Unsupervised analysis of fMRI data using kernel canonical correlation. *Neuroimage*, 37(4):1250–1259, 2007.

T. Hastie, R. Tibshirani, and A. Buja. Flexible discriminant and mixture models. In J. Kay and D. Titterington, editors, *Neural Networks and Statistics*. Oxford University Press, Oxford, 1995.

G. E. Hinton. Products of experts. In *Proceedings of the Ninth International Conference on Artificial Neural Networks*, volume 1, pages 1–6, 1999.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–41,498–520, 1933.

H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.

W. Hsieh. Nonlinear canonical correlation analysis by neural networks. *Neural Networks*, 13:1095–1105, 2000.

W. Hsieh. Nonlinear canonical correlation analysis of the tropical pacific climate variability using a neural network approach. *J. Climate*, 14:2528–2539, 2001.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley, New York, NY, 2001.

A. T. Ihler, J. W. Fisher, and A. S. Willsky. Nonparametric hypothesis tests for statistical dependency. *IEEE Transactions on Signal Processing*, 52(8):2234–2249, 2004.

S. Jain and R. Neal. A split-merge Markov chain procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13:158–182, 2004.

S. Jain and R. M. Neal. Splitting and merging components of a nonconjugate Dirichlet process mixture model. *Bayesian Analysis*, 2(3):445–472, 2007.

B. Jensen. Exploratory datamining in music. Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2006.

M. I. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. An introduction to variational methods for graphical models. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 105–162. MIT Press, Cambridge, 1999.

S. Kaski and J. Sinkkonen. Principle of learning metrics for exploratory data analysis. *Journal of VLSI Signal Processing, special issue on Machine Learning for Signal Processing*, 37:177–188, 2004.

S. Kaski, J. Kangas, and T. Kohonen. Bibliography of self-organizing map (SOM) papers: 1981-1997. *Neural Computing Surveys*, 1:102–350, 1998.

S. Kaski, J. Sinkkonen, and J. Peltonen. Bankruptcy analysis with self-organizing maps in learning metrics. *IEEE Transactions on Neural Networks*, 12:936–947, 2001.

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

S. Kaski, J. Nikkilä, J. Sinkkonen, L. Lahti, J. Knuuttila, and C. Roos. Associative clustering for exploring dependencies between functional genomics data sets. *IEEE/ACM Transactions on Computational Biology and Bioinformatics, Special Issue on Machine Learning for Bioinformatics – Part 2*, 2(3):203–216, 2005.

R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90:773–795, 1995.

M. Kendall. A new measure of rank correlation. *Biometrika*, 30:81–89, 1938.

J. Kettenring. Canonical analysis of several sets of variables. *Biometrika*, 58(3):433–451, 1971.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, volume 2, pages 1137–1143, 1992.

T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, 3rd edition, 2001.

S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

S. Kumar, E. B. Martina, and A. J. Morris. Non-linear canonical correlation analysis using a RBF network. In *Proceedings of the 10th European Symposium on Artificial Neural Networks*, pages 507–512, 2002.

K. Kurihara, M. Welling, and N. A. Vlassis. Accelerated variational dirichlet process mixtures. In B. Schölkopf, J. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems 19*, pages 761–768, 2007.

P. L. Lai and C. Fyfe. A neural implementation of canonical correlation analysis. *Neural Networks*, 12:1391–1397, 1999.

G. Leen and C. Fyfe. A Gaussian process latent variable model formulation of canonical correlation analysis. In *Proceedings of 14th European Symposium on Artificial Neural Networks*, pages 418–418, 2006.

Y. Li and J. Shawe-Taylor. Using KCCA for Japanese-English cross-language information retrieval and document classification. *Journal of intelligent information systems*, 27 (2):117–133, 2006.

L.Song, A.Smola, K.Borgwardt, and A.Gretton. Colored maximum variance unfolding. In *Advances in Neural Information Processing Systems 21*, pages 1–8, 2008.

Z. Lu and T. K. Leen. Semi-supervised clustering with pairwise constraints: A discriminative approach. In *Proceedings of 11th International Conference on Artificial Intelligence and Statistics*, 2007a.

Z. Lu and T. K. Leen. Penalized probabilistic clustering. *Neural Computation*, 19:1528–1567, 2007b.

G. J. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, New York, NY, 2000.

T. Melzer. *Generalized canonical correlation analysis for object recognition*. PhD thesis, Vienna technical university, 2002.

R. C. Meyer, M. Steinfath, J. Lisec, M. Becher, H. Witucka-Wall, O. Törjék, O. Fiehn, Änne Eckardt, L. Willmitzer, J. Selbig, and T. Altmann. The metabolic signature related to high plant growth rate in Arabidopsis thaliana. *PNAS*, 104(11):4759–4764, 2007.

R. Neal. Markov chain sampling methods for Dirichlet process mixture models. Technical Report 9815, Department of statistics, University of Toronto, 1998.

A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*. MIT Press, Cambridge, MA, 2002.

J. Nikkilä, C. Roos, E. Savia, and S. Kaski. Explorative modeling of yeast stress response and its regulation with gCCA and associative clustering. *International Journal of Neural Systems*, 15(4):237–246, 2005.

P. Nymark, P. M. Lindholm, M. V. Korpela, L. Lahti, S. Ruosaari, S. Kaski, J. Hollmén, S. Anttila, V. L. Kinnula, and S. Knuutila. Gene expression profiles in asbestos-exposed epithelial and mesothelial lung cell lines. *BMC Genomics*, 8(62), 2007.

M. Oja, S. Kaski, and T. Kohonen. Bibliography of self-organizing map (SOM) papers: 1998-2001 addendum. *Neural Computing Surveys*, 3:1–156, 2003.

L. Paninski. Estimation of entropy and mutual information. *Neural Computation*, 15(6): 1191–1253, 2003.

E. Parkhomenko, D. Tritchler, and J. Beyene. Genome-wide sparse canonical correlation of gene expression with genotypes. *BMC Proceedings*, 1(S119), 2007.

E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

K. Pearson. Mathematical contributions to the theory of evolution III: Regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London*, A187: 253–318, 1896.

J. Peltonen. *Data exploration with learning metrics*. PhD thesis, Helsinki University of Technology, 2004.

J. Peltonen and S. Kaski. Discriminative components of data. *IEEE Transactions on Neural Networks*, 16(1):68–83, 2005.

J. Peltonen, A. Klami, and S. Kaski. Learning more accurate metrics for self-organizing maps. In J. R. Dorronsoro, editor, *Artificial Neural Networks—ICANN 2002*, pages 999–1004. Springer, Berlin, 2002a.

J. Peltonen, J. Sinkkonen, and S. Kaski. Discriminative clustering of text documents. In L. Wang, J. C. Rajapakse, K. Fukushima, S.-Y. Lee, and X. Yao, editors, *Proceedings of ICONIP'02, 9th International Conference on Neural Information Processing*, pages 1956–1960. IEEE, Piscataway, NJ, 2002b.

J. Peltonen, J. Sinkkonen, and S. Kaski. Sequential information bottleneck for finite data. In R. Greiner and D. Schuurmans, editors, *Proceedings of the Twenty-First International Conference on Machine Learning (ICML 2004)*, pages 647–654. Omnipress, Madison, WI, 2004.

J. Peltonen, J. Goldberger, and S. Kaski. Fast semi-supervised discriminative component analysis. In K. Diamantaras, T. Adali, I. Pitas, J. Larsen, T. Papadimitriou, and S. Douglas, editors, *Machine Learning for Signal Processing XVII*, pages 312–317. IEEE, 2007.

F. Pereira, N. Tishby, and L. Lee. Distributional clustering of English words. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 183–190. ACL, Columbus, OH, 1993.

M. Pöllä, T. Honkela, and T. Kohonen. Bibliography of self-organizing map (SOM) papers: 2002-2005. Technical report, Helsinki University of Technology, 2006.

C. Rasmussen. The infinite Gaussian mixture model. In S. A. Solla, T. K. Leen, and K.-R. Muller, editors, *Proceedings of Neural Information Processing Systems*, pages 554–560, 2000.

C. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

A. Rényi. On measures of dependence. *Acta mathematica Academiae scientiarum Hungaricae*, 10:441–451, 1959.

Y. D. Rubinstein and T. Hastie. Discriminative vs informative learning. In D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, editors, *Proc. ACM KDD*, pages 49–53. AAAI Press, 1997.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

S. Rüping and T. Scheffer, editors. *Proceedings of the ICML Workshop on Learning with Multiple Views*, 2005.

J. Salojärvi, S. Kaski, and J. Sinkkonen. Discriminative clustering in Fisher metrics. In O. Kaynak, E. Alpaydin, E. Oja, and L. Xu, editors, *Artificial Neural Networks and Neural Information Processing - Supplementary proceedings ICANN/ICONIP 2003*, pages 161–164. Istanbul, Turkey, June 2003.

J. W. Sammon, Jr. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18:401–409, 1969.

C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

H. Shen, S. Jegelka, and A. Gretton. Fast kernel ICA using an approximate Newton method. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, pages 476–483, 2007.

C. Sigg, B. Fischer, B. Ommer, V. Roth, and J. Buhmann. Nonnegative CCA for audiovisual source separation. In *Proceedings of Machine learning for Signal Processing*, pages 253–258, 2007.

J. Sinkkonen and S. Kaski. Clustering based on conditional distributions in an auxiliary space. *Neural Computation*, 14:217–239, 2002.

J. Sinkkonen, S. Kaski, and J. Nikkilä. Discriminative clustering: Optimal contingency tables by learning metrics. In T. Elomaa, H. Mannila, and H. Toivonen, editors, *Proceedings of the ECML'02, 13th European Conference on Machine Learning*, pages 418–430. Springer, Berlin, 2002.

N. Slonim and N. Tishby. Agglomerative information bottleneck. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 617–623. MIT Press, Cambridge, MA, 2000.

N. Slonim, N. Friedman, and N. Tishby. Unsupervised document classification using sequential information maximization. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 129–136. ACM Press, New York, NY, USA, 2002.

L. Song, A. Smola, K. Borgwardt, and A. Gretton. Colored maximum variance unfolding. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1385–1392. MIT Press, Cambridge, MA, 2008.

C. E. Spearman. Proof and measurement of association between two things. *American Journal of Psychology*, 1904.

M. Steinfath, D. Groth, J. Lisec, and J. Selbig. Metabolite profile analysis: From raw data to regression and classification. *Physiologia Plantarum*, 132(2):150–161, 2008.

N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. In B. Hajek and R. S. Sreenivas, editors, *Proceedings of The 37th Annual Allerton Conference on Communication, Control, and Computing*, pages 368–377. University of Illinois, Urbana, Illinois, 1999.

K. Torkkola. Feature extraction by non-parametric mutual information maximization. *Journal of Machine Learning Research*, 3:1415–1438, 2003.

J. Tukey. *Exploratory data analysis*. Addison-Wesley, 1977.

M. M. Van Hulle. Constrained subspace ICA based on mutual information optimization directly. *Neural Computation*, 20(4):964–973, 2008.

H. Vinod. Canonical ridge and econometrics of joint production. *Journal of Econometrics*, 4(2):147–166, 1976.

C. Wang. Variational Bayesian approach to canonical correlation analysis. *IEEE Transactions on Neural Networks*, 18:905–910, 2007.

Y. Yamanishi, J.-P. Vert, A. Nakaya, and M. Kanehisa. Extraction of correlated gene clusters from multiple genomic data by generalized kernel canonical correlation analysis. *Bioinformatics*, 19:i323–i330, 2003.

Y. Yamanishi, J.-P. Vert, and M. Kanehisa. Protein network inference from multiple genomic data: A supervised approach. *Bioinformatics (Proceedings of ISMB2004)*, 20: 363–370, 2004.

X. Yin. Canonical correlation analysis based on information theory. *Journal of multivariate analysis*, 91:161–176, 2004.

J. Ylipaavalniemi, E. Savia, R. Vigário, and S. Kaski. Functional elements and networks in fMRI. In *Proceedings of the 15th European Symposium on Artificial Neural Networks*, pages 561–566, 2007.