Seppo Virtanen

# Bayesian exponential family projections

**Faculty of Electronics, Communications and Automation**

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 10.5.2010

**Thesis supervisor:**

Prof. Samuel Kaski

**Thesis instructor:**

D.Sc. (Tech.) Arto Klami

Aalto University
School of Science
and Technology

AALTO-YLIOPISTO
TEKNILLINEN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Seppo Virtanen

Työn nimi: Bayesilaisia projektiomenetelmiä eksponentiaaliperheissä

Päivämäärä: 10.5.2010 Kieli: Englanti Sivumäärä:6+47

Elektroniikan, tietoliikenteen ja automaation tiedekunta

Professuuri: Informaatiotekniikka Koodi: T-61

Valvoja: Prof. Samuel Kaski

Ohjaaja: TkT Arto Klami

Eksploratiivinen data-analyysi tarkoittaa oleellisen informaation löytämistä tietoaineistoista. Koneoppimismenetelmät automatisoivat tämän tavoitteen sovittamalla dataan malleja. On oleellista, että kaikki taustatieto voidaan käyttää kyseisten mallien rakentamiseen.

Pääkomponenttianalyysi on tyypillinen koneoppimismenetelmä eksploratiiviseen analyysiin. Viime aikoina sen probabilistiset tulkinnat ovat osoittaneet menetelmän rajoittuneisuuden tietyn tyyppiseen dataan. Pääkomponenttianalyysin laajennus eksponentiaaliperheen jakaumiin korjaa tämän ongelman.

Työssä esitetään yleinen malliperhe, joka soveltuu usean aineiston analyysiin, rakentamalla pääkomponenttianalyysin eksponentiaaliperheen laajennuksen päälle. Yhtenäinen viitekehys sisältää menetelmiä, jotka soveltuvat ohjattuun ja ohjaamattomaan oppimiseen.

Aiemmista menetelmistä poiketen työssä käytetään Bayesilaista menetelmää suurimman uskottavuuden menetelmän sijaan. Bayesilaisessa menetelmässä taustatietoa voidaan esittää priorijakaumien muodossa. Työssä esitetään yleinen priorijakauma, jolla voidaan ottaa jakaumille tyypilliset piirteet huomioon.

Työssä esitetään useita parannuksia mallintamiseen, mallien rakentamiseen, oppimiseen ja tulkintaan liittyen. Empiirisillä kokeilla osoitetaan, että esitetyt menetelmät toimivat paremmin kuin perinteiset menetelmät.

Avainsanat: approksimatiivinen Bayesilainen inferenssi, Bayesilainen mallintaminen, eksponentiaaliperhe, kanoninen korrelaatioanalyysi, ohjaamaton ja ohjattu oppiminen, pääkomponenttianalyysi

AALTO UNIVERSITY
SCHOOL OF SCIENCE AND TECHNOLOGY

ABSTRACT OF THE
MASTER'S THESIS

Author: Seppo Virtanen

Title: Bayesian exponential family projections

Date: 10.5.2010 Language: English Number of pages:6+47

Faculty of Electronics, Communications and Automation

Professorship: Information technology Code: T-61

Supervisor: Prof. Samuel Kaski

Instructor: D.Sc. (Tech.) Arto Klami

Exploratory data analysis stands for extracting useful information from data sets. Machine learning methods automate this process by fitting models to data. It is essential to provide all available background knowledge for building such models.

Principal component analysis is a standard method for exploratory data analysis. Recently its probabilistic interpretation has illustrated that it is only suitable for a specific type of data. Extension of principal component analysis to the exponential family removes this problem.

In this thesis a general model family suitable for the analysis of multiple data sources is presented by building on the exponential family principal component analysis. The unifying framework contains as special cases methods suitable for unsupervised and supervised learning.

While earlier methods have mainly relied on maximum likelihood inference, in this thesis Bayesian modeling is chosen. In Bayesian modeling background knowledge is utilized in the form of prior distributions. In this thesis, a general prior distribution is proposed that takes distribution-specific constraints into account.

Multiple contributions to modeling, inference and model interpretation are introduced. With empirical experiments it is demonstrated how the proposed methods outperform traditional methods.

Keywords: approximative Bayesian inference, Bayesian modeling, canonical correlation analysis, exponential family, principal component analysis, supervised and unsupervised learning

# Preface

The work was carried out in the Helsinki Institute for Information Technology HIIT, Department of Information and Computer Science, funded by the Adaptive Informatics Research Centre at the Aalto University School of Science and Technology. My research was in part to develop methods for the aivoAALTO research project.

I thank my supervisor Prof. Samuel Kaski and instructor D.Sc. Arto Klami for the supervision and guidance. Additionally, I thank the ICS department for providing excellent facilities for the research and the members of the MI-group for providing an academic environment to work in.

Otaniemi, 10.5.2010

Seppo Virtanen

# Contents

# 1 Introduction

Modern computer science enables storing and processing large collections of data. Such data collections include for example image data, gene expression measurements or functional resonance imaging (fMRI) data, to name a few, with applications ranging from image restoration to prediction of human brain activation patterns for natural stimuli. Other exemplary application is movie recommendation system, where the common task is to predict missing items based on observed relations. This application is used throughout the introduction as an illustrating example. For all those applications data analysis is needed for finding useful information.

Given data of movie ratings of users the aim is to build a system that can recommend movies. Of course, the users do not rate movies randomly, but instead there is some process behind the data generation. The task in data analysis is to uncover this process; however, the true data generating process may be too complex. In practice, it suffices to make accurate predictions, hence good and useful approximative models of reality are considered.

Machine learning aims to build models that learn from data. It is based on mathematical models that are designed for different tasks by making sets of assumptions. Models have parameters which are fitted to data based on some criterion; this process is also termed learning. After the model parameters are fitted, the model can be used, for example, to make predictions and explain the data. The model has learned relevant structure from the data if it can be used to explain the observed data and to make good predictions. For example, an approximative model for the data generating process can be used to predict future data and impute missing values.

Principal component analysis (PCA) is a traditional, over a century old, machine learning method suitable for finding structure in a data set (Jolliffe, 1986). The $N \times D$ data matrix is denoted with $\mathbf{X}$. In the movie recommendation application the $N$ rows of $\mathbf{X}$ correspond to the different users and the $D$ columns correspond to the different movies. The data matrix element $x_{nd}$ is the rating of user $n$ for movie $d$. The model for PCA can be written as

$$\mathbf{X} \;=\; \mathbf{U}\mathbf{V}^T \;+\; \mathbf{E} \quad \text{or} \quad x_{nd} = \textstyle\sum_{k=1}^{K} u_{nk} v_{dk} + \epsilon_{nd}, \tag{1}$$

where the $N \times K$ matrix $\mathbf{U}$ is a row-wise collection of latent variables assigned for each data point, $D \times K$ matrix $\mathbf{V}$ is a projection from latent variables to data, and $\mathbf{E}$ is a noise matrix. The $K$ is the rank of the decomposition. Essentially, PCA searches for two matrices $\mathbf{U}$ and $\mathbf{V}$ that capture the relevant properties of the data. This task is termed dimensionality reduction: describing the observed high-dimensional data with fewer features assuming that $K$ is much smaller than $D$. After finding the parameters $\mathbf{U}$ and $\mathbf{V}$, the missing items can be predicted or $\mathbf{U}$ can be analyzed, for example, to see if users form groups.

Probabilistic modeling is one way of formulating models such as PCA. Data is connected to the parameters through the *likelihood function* that represents conditional probability of data given the parameters. Most common inference methods for probabilistic modeling can be divided to two different approaches, Bayesian and

maximum likelihood (ML) inference methods. Maximum likelihood seeks parameter values that are most probable measured by the likelihood function. In Bayesian inference, first a full joint probability model for data and parameters is built by setting a prior distribution for the parameters. The parameters are then conditioned on observed data; what is the distribution for the parameters after seeing the data.

Probabilistic interpretation of PCA provided by Tipping and Bishop (1999) shows that PCA is only optimal for a specific type of data. Essentially, PCA assumes that the noise, elements of $\mathbf{E}$, and the latent variables follow Gaussian distribution. The assumption for the noise is ultimately rather restricting.

In recent years one of the main directions in PCA extensions has been to relax the Gaussianity assumption, to better suit domains with non-continuous-valued data. The movie rating matrix is a suitable example as the entries of the data matrix are discrete and typically range from $1-5$. For such data the 'measurement noise' is not Gaussian. A true rating of 4 might correspond to 3 or 5 but definitely not, for example, $-0.52$ or 6.98 that would be possible for the Gaussian noise. In addition to ordinal data type, binary and integer data types also have practical applications. For example, documents can be represented by binary features that indicate whether a certain word appeared in the document, or by counts that tell how many times the word appeared in the document. All the above discussed data types have an interesting common property: they belong to the so-called exponential family.

The first exponential family variant of PCA (EPCA; Collins et al., 2002) introduced the basic approach of taking the data distribution into account. Exponential family is a collection of different probability distributions that share the same functional form (see Bernardo and Smith (2000)).

EPCA still remains an active research area. Examples of recent advances in EPCA family of models include a semi-parametric formulation applicable to even more flexible distributions (Sajama and Orlitsky, 2004) and more efficient algorithms guaranteed to converge to the global optimum (Guo and Schuurmans, 2008). Most of the presented methods are limited to maximum likelihood inference, however, Bayesian exponential family PCA (BEPCA) takes the approach to the next level by including a full probability model for the data and the parameters. Mohamed et al. (2009) made a straightforward assumption of Gaussian priors for the latent variables.

In an abstract and compact form, the PCA problem is simply a matrix decomposition. The EPCA makes the decomposition in the space of the so-called natural parameters of element-wise exponential family distributions. That is, each element of $\mathbf{X}$ is assumed to be generated independently from an exponential family distribution with parameters collected into $\mathbf{\Theta}$, while $\mathbf{\Theta}$ itself is factorized as $\mathbf{\Theta} = \mathbf{U}\mathbf{V}^T$.

While EPCA focuses on decomposing a single data matrix, additional data about users or movies could be used to improve recommendation accuracy. Supervised PCA (both exponential family and standard; Yu et al. 2006; Guo 2009) are the simplest generalizations of PCA suitable for the analysis of multiple data sets. Category labels are a special case of additional data that represent especially interesting properties. Instead of recommending movies to users it may be more interesting to predict how well the movie is going to sell. Providing PCA with such label infor-

mation results in supervised projections.

## 1.1 Contributions and contents

In this thesis two novel extensions of EPCA are presented for the analysis of two (or more) co-occurring data sets, namely exponential family partial least squares (EPLS) and exponential family canonical correlation analysis (ECCA). Let $\mathbf{Y}_1$ and $\mathbf{Y}_2$ denote two data sets with dimensionalities $N \times D_1$ and $N \times D_2$. The samples co-occur, meaning that the rows of $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are paired. EPLS is used for prediction tasks. For example, treating $\mathbf{Y}_1$ as label-information, it separates variation that is shared between $\mathbf{Y}_1$ and $\mathbf{Y}_2$ from variation that is specific for $\mathbf{Y}_2$. Motivation is that not all variation in $\mathbf{Y}_2$ is relevant for predicting $\mathbf{Y}_1$. While EPLS focuses on prediction, ECCA can be used to find what is shared between the data sources. Shared variation between $\mathbf{Y}_1$ and $\mathbf{Y}_2$ is captured by discarding set-specific aspects. More intuitively, ECCA can be seen as a data fusion method assuming that only commonalities between the two data sets are interesting.

It is demonstrated in this thesis how Bayesian exponential family variants of supervised EPCA (Guo, 2009), partial least squares (PLS), and canonical correlation analysis (CCA) can be obtained using the same basic formulation by providing a unifying framework. The proposed methods extend naturally the recent literature on probabilistic variants of these methods (PLS: Gustafsson, 2001; Nounou et al., 2002, CCA: Bach and Jordan, 2005; Klami and Kaski, 2007), in the same way as the EPCA approaches build on top of probabilistic PCA.

In Bayesian modeling prior distributions need to be assigned for the parameters of the model. In this thesis, ways of postulating priors for $(\mathbf{U}, \mathbf{V})$ and for computing with them are introduced. In general, the domain of natural parameters is constrained. Assuming a Gaussian prior for the latent variables is not suitable, for example, for exponential distribution where the natural parameters are restricted to be positive. In this thesis, a novel regularizing prior is introduced, that removes some of the problems of the Gaussianity assumption by constraining the values for the $\mathbf{\Theta}$.

This thesis is structured as follows. In Section 2, probabilistic modeling is discussed introducing elementary Bayesian inference in more detail. In Section 3, first exponential family distributions and some of their central properties are reviewed. Secondly, standard PCA and its generalization to the exponential family are presented. One of the main contributions of this thesis is then presented in Section 4, where the assumptions that result in methods suitable for the interesting case of two-view analysis of co-occurring data sources are introduced in detail. Then in Section 5 novel ways of defining suitable priors for the models are presented, and efficient inference algorithms are presented in Section 6. Finally in Section 7, the models are demonstrated to outperform their rivals in a number of experiments using both artificial and real data. Discussion is given in Section 8.

# 2 Modeling background

Probabilistic models describe the generation of data by probability distributions. A parametric generative probabilistic model defines a probability distribution

$$p(\mathbf{X}|\mathbf{\Theta}),$$

where $\mathbf{X}$ denotes observed data and $\mathbf{\Theta}$ is the collection of model parameters.

## 2.1 Bayesian inference

In Bayesian inference joint probability distribution is defined for observed and unobserved quantities. This can be written applying the conditional probability formula as

$$p(\mathbf{X}, \mathbf{\Theta}) = p(\mathbf{X}|\mathbf{\Theta})p(\mathbf{\Theta}),$$

where $p(\mathbf{\Theta})$ is the prior distribution for parameters denoted with $\mathbf{\Theta}$, while $p(\mathbf{X}|\mathbf{\Theta})$ is the likelihood function. The likelihood function essentially measures how probable it is to observe $\mathbf{X}$ if $\mathbf{\Theta}$ are the parameters. Applying the conditional probability formula one more time, the posterior distribution of the parameters is

$$p(\mathbf{\Theta}|\mathbf{X}) = \frac{p(\mathbf{X}|\mathbf{\Theta})p(\mathbf{\Theta})}{p(\mathbf{X})},$$

where the normalization term $p(\mathbf{X})$ is used to ensure that the posterior distribution is valid, i.e., integrates to one over the whole parameter space. The posterior distribution measures how probable the values for the parameters are *after* seeing the data $\mathbf{X}$. This is the core of Bayesian inference, to condition parameters on observed data. Instead of learning specific optimal values for the variables as done in optimization-based learning frameworks , the Bayesian inference process considers the full posterior distribution of these variables.

Many common probability distributions can be represented through summary statistics. Mean value is one such quantity and it represents the expected value of the distribution. The expectation of $\mathbf{\Theta}$ (i.e. the mean) with respect to $p(\mathbf{\Theta}|\mathbf{X})$ is defined as

$$\mathbb{E}_{p(\mathbf{\Theta}|\mathbf{X})}[\mathbf{\Theta}] = \int \mathbf{\Theta} p(\mathbf{\Theta}|\mathbf{X}) d\mathbf{\Theta}.$$

When there is no risk of confusion, the subscript of $\mathbb{E}[\cdot]$ is dropped and assumed to be the posterior distribution.

In order to express the posterior distribution analytically, the normalization constant needs to be solved. The normalization term can be written, introducing the relevant concept of *marginalization*, as

$$\begin{aligned} p(\mathbf{X}) &= \int p(\mathbf{X}, \mathbf{\Theta}) d\mathbf{\Theta} \\ &= \int p(\mathbf{X}|\mathbf{\Theta})p(\mathbf{\Theta}) d\mathbf{\Theta}. \end{aligned} \tag{2}$$

In machine learning prediction of new samples is of ultimate interest. The probability for unobserved new data, $\mathbf{x}^*$, is

$$p(\mathbf{x}^*|\mathbf{X}) = \int p(\mathbf{x}^*|\mathbf{\Theta})p(\mathbf{\Theta}|\mathbf{X})d\mathbf{\Theta}.$$

The distribution is called posterior predictive distribution. It can be used to generate new data. For example, the posterior predictive distribution is used to impute the missing values in probabilistic matrix factorization. Integration over the parameter space for prediction results in optimal predictions as predictions based on multiple different models are averaged using the correct weights as determined by the posterior.

## 2.2 Point estimates

In the previous section the full Bayesian inference was discussed. However, this is not the only option, for quick inference point estimates of the posterior distribution can be sought. The most likely parameter values are given by the maximum a posteriori estimate

$$\mathbf{\Theta}_{MAP} = \arg\max_{\mathbf{\Theta}} p(\mathbf{\Theta}|\mathbf{X}) = \arg\max_{\mathbf{\Theta}} p(\mathbf{X}|\mathbf{\Theta})p(\mathbf{\Theta}) \tag{3}$$

by noting that the normalization term does not depend on $\mathbf{\Theta}$. The resulting estimate, $\mathbf{\Theta}_{MAP}$, is the best one if only one has to be chosen. For computational simplicity the logarithm is usually applied to the cost function (3). Logarithm is a monotonic function and does not change the value of $\mathbf{\Theta}_{MAP}$, and the log-posterior is written as

$$\mathcal{L}_{MAP} = \ln p(\mathbf{X}|\mathbf{\Theta}) + \ln p(\mathbf{\Theta}).$$

Unfortunately, the use of $\mathbf{\Theta}_{MAP}$ does not reveal the uncertainty of the posterior distribution. The predictive distribution given a point estimate is

$$p(\mathbf{x}^*|\mathbf{X}) = p(\mathbf{x}^*|\mathbf{\Theta}_{MAP}),$$

resulting in predictions that are best possible ones given just one value for $\mathbf{\Theta}$, yet suboptimal compared to averaging over the whole posterior.

Computationally similar method to MAP is so-called maximum likelihood inference. In maximum likelihood the prior distribution $p(\mathbf{\Theta})$ that is used to constrain or regularize the space of possible solutions is omitted. To justify this approach from Bayesian point of view the prior is set uniform, that is, $p(\mathbf{\Theta})$ is constant.

Maximum likelihood inference for $\mathbf{\Theta}$ can be written as

$$\mathbf{\Theta}_{ML} = \arg\max_{\mathbf{\Theta}} p(\mathbf{X}|\mathbf{\Theta}).$$

or in the log-domain as

$$\mathbf{\Theta}_{ML} = \arg\max_{\mathbf{\Theta}} \mathcal{L} = \arg\max_{\mathbf{\Theta}} \ln p(\mathbf{X}|\mathbf{\Theta}). \tag{4}$$

## 2.3 Approximate Bayesian inference

While the point estimates can be searched by optimization, the Bayesian inference requires integration over the parameter space. In order to obtain the posterior distribution, it is necessary to marginalize over $\boldsymbol{\Theta}$. At this point the seemingly simple expression of Bayes formula turns out to be a rather tedious one. For many non-trivial models the integral cannot be expressed in closed form resulting in a posterior distribution of any known form. This does not prevent, however, from using the model, since the marginal distribution and other relevant quantities can be solved with approximative inference methods.

There are two common approximation methods to solve the complicated integrals. One approach is to approximate the distributions such that exact integration is viable (Bishop, 2006). This approach is called variational Bayes. In this thesis, however, this kind of approximations are not discussed further. Instead, the focus is on sampling methods (Gelman et al., 2004). The reason is that sampling methods require minor changes in computations when the model structure is changed. For variational techniques even minor modifications lead to elaborate computations. However, sampling methods require usually considerably more computation time than variational methods.

As mentioned in the previous section, the posterior predictive distribution is defined as the integral

$$p(\mathbf{x}^*|\mathbf{X}) = \int p(\mathbf{x}^*|\boldsymbol{\Theta})p(\boldsymbol{\Theta}|\mathbf{X})d\boldsymbol{\Theta}. \tag{5}$$

The idea in sampling methods is to approximate this with

$$p(\mathbf{x}^*|\mathbf{X}) = \frac{1}{S}\sum_{s=1}^{S} p(\mathbf{x}^*|\boldsymbol{\Theta}^{(s)}),$$

where $\boldsymbol{\Theta}^{(s)} \sim p(\boldsymbol{\Theta}|\mathbf{X})$ with $s = 1, \ldots, S$ are samples from the posterior. The approach is possible because samples $\boldsymbol{\Theta}^{(s)}$ can be drawn even when $p(\mathbf{X})$ is unknown.

Markov Chain Monte Carlo (MCMC) is an umbrella term for a myriad of methods that can be used to obtain samples from the posterior. The basis of MCMC is the Metropolis-Hastings (MH) algorithm (Gelman et al., 2004). The sampler is conceptually simple; it proceeds by proposing a shift to the current state $\boldsymbol{\Theta}^c$ from the (symmetric) proposal distribution $q(\cdot)$. Proposal distributions are typically specified separately for different variables in $\boldsymbol{\Theta}$. The proposed state, $\boldsymbol{\Theta}^*$, is drawn from $q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^c)$ and accepted with probability

$$\min\big(1, \frac{p(\boldsymbol{\Theta}^*|\mathbf{X})q(\boldsymbol{\Theta}^c|\boldsymbol{\Theta}^*)}{p(\boldsymbol{\Theta}^c|\mathbf{X})q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}^c)}\big). \tag{6}$$

If the proposal is rejected the state does not change. The acceptance probability simplifies to

$$\min\big(1, \frac{p(\boldsymbol{\Theta}^*|\mathbf{X})}{p(\boldsymbol{\Theta}^c|\mathbf{X})}\big)$$

for symmetric proposal distributions, $q(\boldsymbol{\Theta}^c|\boldsymbol{\Theta}^*) = q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}^c)$. Computing the acceptance probability is possible because the normalization term cancels out:

$$\frac{p(\boldsymbol{\Theta}^*|\mathbf{X})}{p(\boldsymbol{\Theta}^c|\mathbf{X})} = \frac{p(\mathbf{X}|\boldsymbol{\Theta}^*)p(\boldsymbol{\Theta}^*)p(\mathbf{X})}{p(\mathbf{X}|\boldsymbol{\Theta}^c)p(\boldsymbol{\Theta}^c)p(\mathbf{X})} = \frac{p(\mathbf{X}|\boldsymbol{\Theta}^*)p(\boldsymbol{\Theta}^*)}{p(\mathbf{X}|\boldsymbol{\Theta}^c)p(\boldsymbol{\Theta}^c)}.$$

Starting from some initial value of $\boldsymbol{\Theta}$ and proposing infinitely many proposals the method ultimately provides samples from the posterior distribution; when that happens the sampler is said to be converged. However, it is not trivial to determine convergence. Gelman et al. (2004) propose using a method they call the potential scale reduction factor (PSRF) to assess convergence.

Gibbs sampling is another common MCMC method for Bayesian inference. It is a special case of the MH algorithm. The conditional distributions of the parameters are used for proposals. The conditional distribution for $\boldsymbol{\Theta}_i$ is defined as $p(\boldsymbol{\Theta}_i|\mathbf{X}, \boldsymbol{\Theta}_{-i})$, where $\boldsymbol{\Theta}_{-i}$ denotes the set of all other parameters except $\boldsymbol{\Theta}_i$. The method proceeds by updating the parameters sequentially, proposing for each a new value using the corresponding conditional distribution. Rejections do not occur because of using conditional distributions (Gelman et al., 2004); the acceptance probability is always one.

Typically, in modeling only a few parameters are of interest. The parameters that are necessary for the model but not for the further analysis are called nuisance parameters. Hence, the marginal posterior distributions, for instance, $p(\boldsymbol{\Theta}_i|\mathbf{X})$ are interesting. In sampling, marginalization of $\boldsymbol{\Theta}_{-i}$ can be performed by sampling all of the parameters from the joint model, and simply discarding the values for $\boldsymbol{\Theta}_{-i}$.

## 2.4 Model selection

Point estimation methods rely on using a single model instead of averaging over multiple models as in Bayesian inference. The problem of model selection is choosing one model from many possible alternative models. The decision can be done by choosing the model that generalizes well to new data.

The training error is denoted as $J_{train}$ and the error for future data as $J_{test}$. Demonstration of the model selection procedure is given in Figure 1. The model complexity (usually measured by the number of the parameters in the model) needs to be set suitably. Too complex model overfits, that is, it describes well training data but does not generalize well. On the other hand, the model has underfitted if both errors are large.

In principle, model complexity could be chosen based on $J_{test}$ but this quantity is not known. By using a validation set we can approximate $J_{test}$ and determine model complexity (Bishop, 2006). The available data are split in two sets and one is used for training and the other for validation. When the model begins to describe aspects of training data alone the prediction error increases or remains the same. For MAP estimation the validation set can be used to set the parameters of the prior as well.

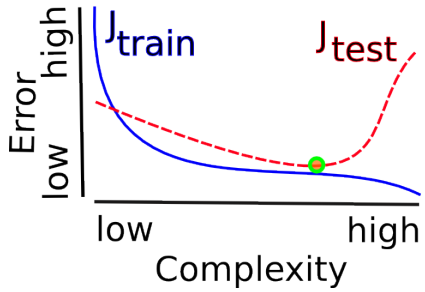



Figure 1: Demonstration of model selection. Low complexity models underfit and both training and testing errors are large. On the other hand, too complex models overfit and the test error is large. Suitable compromise of model complexity results in best performance.

## 2.5 Why Bayesian modeling?

The difference between maximum likelihood(ML) and Bayesian inference methods is critical. For Bayesian inference the uncertainty is expressed in parameter values in the form of prior distribution and the corresponding posterior distribution is sought. ML assumes that the observed data set has unique true parameters and the uncertainty is assumed only for the observed data, i.e., the observed data set is one random realization of the true process. The solutions found by ML lie in the space spanned by the observed data, making this approach well justified for large data sets. For the Bayesian method the space of parameters is constrained by the prior distribution. For example, in extreme cases the parameter value $\mathbf{\Theta}_0$ with zero prior probability, $p(\mathbf{\Theta}_0) = 0$, results in posterior with zero probability, $p(\mathbf{X}|\mathbf{\Theta}_0)p(\mathbf{\Theta}_0) = 0$.

Predictive quality of ML for machine learning methods depends on the number of training samples. Simple illustration of this effect is demonstrated by classical coin tossing example. If a coin has been tossed two times and both tosses landed heads, ML deducts that all future tosses are also heads. ML is overly confident in its predictions. In Bayesian inference a prior favoring a fair coin is used, and the posterior distribution after the two tosses contains the prior knowledge resulting in rational inference for the future outcomes.

Briefly put, ML overfits to small data sets while computationally heavy Bayesian methods flourish. However, this does not mean that Bayesian inference would be limited to only small data sets. For example, Salakhutdinov and Mnih (2008 (b)). apply Bayesian matrix factorization to a very large matrix.

# 3 Exponential family projection models

In the previous section the basic concepts of Bayesian modeling were discussed. In this section concrete models are introduced: Principal component analysis (PCA) and its extension to the exponential family. The section starts with explanation of the exponential family and then the PCA and its extensions are explained. In the end of this section, a brief review on related methods is given. The focus in this section is on ML inference, whereas Bayesian inference solutions for this kind of models are presented in Section 6.2.

## 3.1 Exponential family distribution

Exponential family is a collection of distributions that can be used to approximate all relevant and common distributions usually encountered in machine learning and modeling in general (Bernardo and Smith, 2000; Bishop, 2006; Gelman et al., 2004).

A univariate random variable $x \in \mathcal{X} \subseteq \mathbb{R}$ (where $\mathcal{X}$ is a suitable subset of the real-space, such as $\mathbb{Z}$ or $\mathbb{R}_+$) in the exponential family follows the distribution

$$p(x|\theta) = \exp(s(x)\theta + \ln h(x) - g(\theta)), \tag{7}$$

where $\theta \in \mathbb{K} \subseteq \mathbb{R}$ represents the natural parameters of the distribution, $g(\cdot)$ is the log cumulant function that normalizes $p(x|\theta)$ to be a valid distribution, $s(\cdot)$ are the sufficient statistics, and $h(\cdot)$ is a function of the data alone.

To be more specific, (7) can be written as

$$p(x|\theta) = \frac{h(x)\exp(s(x)\theta)}{\int_{x\in\mathcal{X}} h(x)\exp(s(x)\theta)dx},$$

where

$$g(\theta) = \ln \int_{x\in\mathcal{X}} h(x)\exp(s(x)\theta)dx$$

is the normalization term. In this thesis distributions are confined to the natural exponential family by assuming $s(x) = x$. An additional assumption is that the exponential family is regular, that is, the function $h(\cdot)$ does not depend on $\theta$. Otherwise the family is said to be non-regular. Different choices of $g(\cdot)$ lead to different exponential family distributions including Gaussian with known variance, Bernoulli, Poisson, and exponential. The functional form of $g(\cdot)$ depends on the data domain $\mathcal{X}$ and $h(\cdot)$.

The function $g(\theta)$ has interesting properties. By differentiating it with respect

to the parameters one obtains

$$\begin{aligned}\frac{d}{d\theta}g(\theta) &= g'(\theta) \\ &= \frac{d}{d\theta}\ln\int_{x\in\mathcal{X}} h(x)\exp(x\theta)dx \\ &= \frac{\int_{x\in\mathcal{X}} xh(x)\exp(x\theta)dx}{\int_{x\in\mathcal{X}} h(x)\exp(x\theta)dx} \\ &= \int_{x\in\mathcal{X}} xh(x)\exp(x\theta - g(\theta))dx \\ &= \mathbb{E}_{p(x|\theta)}[x] = \mu.\end{aligned}$$

That is, the derivative of $g(\theta)$ defines the expectation of $x$. Similarly the $n$th order cumulants can be calculated by differentiating $g(\theta)$ $n$ times.

Exponential family distributions can also be expressed in an alternative parametrization. Above the natural parametrization was presented, while the mean value parametrization, $p(x|\mu)$, is more commonly known. Some examples of the exponential family distributions are collected in Table 1 presenting details of the different parametrizations and of the domains of the data and the parameter. For example, it can be seen that for the Gaussian data the two different parameterizations are equivalent.

As a concrete example, for $\mathcal{X} = \{0, 1\}$ and $h(x) = 1$ the log cumulant function can be written by replacing integration by summation as

$$g(\theta) = \ln(1 + \exp(\theta)).$$

These assumptions lead to the Bernoulli distribution that belongs to the exponential family. Using the identity $\exp(\ln f(x)) = f(x)$ and writing the logarithm of Bernoulli density function one obtains

$$\ln p(x|\mu) = x\ln\mu + (1-x)\ln(1-\mu) = x\ln\frac{\mu}{1-\mu} + \ln(1-\mu) \tag{8}$$

and parameterizes

$$\theta = \ln\frac{\mu}{1-\mu} \tag{9}$$

to obtain inverse mapping

$$\mu = \frac{1}{1+\exp(-\theta)}. \tag{10}$$

Finally inserting (9) and (10) to (8) gives (7) with $\ln h(x) = 0$.

### 3.1.1 Conjugate priors

Exponential families have many interesting properties. One property is that for every member of exponential family there exists a so-called conjugate prior distribution for $\theta$:

$$p(\theta) \propto \exp\big(\lambda\theta - \nu g(\theta)\big). \tag{11}$$

Table 1: Examples of distributions in the exponential family. Symbol $dom()$ is used to denote the domain of the argument. The derivative of $g(\cdot)$ is the so called link function that is needed to transform natural parameter to the data space. In Section 3.3.1 it is shown how the link function arises naturally in exponential family projections.

| | Gaussian, $\sigma^2 = 1$ | Bernoulli | Poisson | Exponential |
|---|---|---|---|---|
| $p(x\|\mu)$ | $\frac{\exp(-1/2(x-\mu)^2)}{\sqrt{2\pi}}$ | $\mu^x(1-\mu)^{1-x}$ | $\exp(-\mu)\frac{\mu^x}{x!}$ | $\mu\exp(-\mu x)$ |
| $dom(x)$ | $\mathbb{R}$ | $\{0,1\}$ | $\{0,1,2,\dots\}$ | $\mathbb{R}_+$ |
| $dom(\mu)$ | $\mathbb{R}$ | $[0,1]$ | $\mathbb{R}_{++}$ | $\mathbb{R}_{++}$ |
| $\theta$ | $\mu$ | $\ln\frac{\mu}{1-\mu}$ | $\ln\mu$ | $-\mu$ |
| $dom(\theta)$ | $\mathbb{R}$ | $\mathbb{R}$ | $\mathbb{R}$ | $\mathbb{R}_{--}$ |
| $g(\theta)$ | $\frac{1}{2}\theta^2$ | $\ln(1+\exp(\theta))$ | $\exp(\theta)$ | $-\ln(-\theta)$ |
| $g'(\theta)$ | $\theta$ | $(1+\exp(-\theta))^{-1}$ | $\exp(\theta)$ | $-\theta^{-1}$ |
| $\ln h(x)$ | $-\frac{1}{2}(x^2+\ln 2\pi)$ | 0 | $-\ln x!$ | 0 |

A prior is defined to be conjugate if the corresponding posterior distribution is of the same form as the prior. The posterior distribution can be written

$$p(\theta|x) \propto p(x|\theta)p(\theta) = \exp\big((x+\lambda)\theta - (1+\nu)g(\theta)\big)$$

and it is indeed of the same form as (11). The main motivation for using such priors is that the posterior distributions can be derived analytically. The prior parameters control the strength of the prior. Values near 0 for $\lambda$ and $\nu$ correspond to weakly informative prior, resulting to $p(\theta|x) \propto p(x|\theta)$. The corresponding conjugate priors for some of the distributions in Table 1 are presented in Table 2.

Table 2: The conjugate priors with alternative parametrization. For example, Bernoulli-beta denotes that the beta distribution is the conjugate prior for the Bernoulli distribution. The normalization term $Z(\cdot)$, as $g(\cdot)$ in (7), depends on the values of the parameters of the distribution.

| | Gaussian-Gaussian | Bernoulli-beta | Poisson-gamma |
|---|---|---|---|
| $p(\mu)$ | $\frac{1}{Z(\mu_0,\sigma^2)}\exp(-\frac{1}{2\sigma^2}(\mu-\mu_0)^2)$<br>$\mu_0 \in \mathbb{R},\ \sigma^2 > 0$ | $\frac{1}{Z(\alpha,\beta)}\mu^\alpha(1-\mu)^\beta$<br>$\alpha,\beta > 0,\ 0 < \mu < 1$ | $\frac{1}{Z(\alpha,\beta)}\mu^{\alpha-1}\exp(-\mu/\beta)$<br>$\alpha,\beta > 0,\ \mu \geq 0$ |
| $\lambda$ | $\mu_0/\sigma^2$ | $\alpha$ | $\alpha - 1$ |
| $\nu$ | $1/(2\sigma^2)$ | $\beta + \alpha$ | $1/\beta$ |

## 3.2 Principal component analysis

As described briefly in Section 1, PCA is a frequently used dimensionality reduction method (See Jolliffe, 1986). The purpose of PCA is to find a low-dimensional representation of data that can be used, for example, for data compression or visualization.

There are two common ways to derive PCA. The first is possibly more common, while the latter provides a deeper understanding of the method. In the first way the PCA can be seen as a method that seeks for projections that capture maximal variance in the projected space (Hotelling, 1933). The other way is to search for low-rank structure of the data by minimizing the reconstruction error between the low-rank approximation and the original data (Pearson, 1901). Below both approaches are presented, omitting unnecessary details that can be found from any reasonable textbook account on machine learning, such as (Bishop, 2006).

For the remainder of this thesis it is assumed that $N$ observed realizations of $D$-dimensional random variable $\mathbf{x} \in \mathbb{R}^D$ are collected in the matrix

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_N \end{pmatrix}^T \in \mathbb{R}^{N \times D}.$$

PCA seeks components, $\mathbf{v}_i \in \mathbb{R}^D$, $i = 1, \dots, K$ that capture maximal variance of the projected data $\mathbf{v}^T\mathbf{x}$ under the constraint that different components are orthogonal, that is, $\mathbf{v}_i^T\mathbf{v}_j = \mathbf{I}_{ij}$, where $\mathbf{I}$ is a $K \times K$ identity matrix. The components are demonstrated in Figure 2 for simulated data. The components correspond to the first $K$ leading eigenvectors of the sample covariance matrix

$$\mathbf{C} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \hat{\boldsymbol{\mu}})(\mathbf{x}_n - \hat{\boldsymbol{\mu}})^T,$$

where $\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$ is the sample data mean. The eigenvectors of $\mathbf{C}$, denoted by the columns of $\widetilde{\mathbf{W}} \in \mathbb{R}^{D \times D}$, correspond to the solutions of the linear system, $\mathbf{C}\widetilde{\mathbf{W}} = \widetilde{\mathbf{W}}\boldsymbol{\Lambda}$, where $\boldsymbol{\Lambda}$ is a diagonal matrix with eigenvalues $\lambda_i$, $i = 1, \dots, D$, on its diagonal sorted as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D$, and $\widetilde{\mathbf{W}}$ is an orthonormal matrix satisfying $\widetilde{\mathbf{W}}^T\widetilde{\mathbf{W}} = \widetilde{\mathbf{W}}\widetilde{\mathbf{W}}^T = \mathbf{I}$.

Choosing the $K$ first eigenvectors of $\widetilde{\mathbf{W}}$ in the columns of $\mathbf{V} \in \mathbb{R}^{D \times K}$ the projection of data to the so-called latent variables is defined as

$$\mathbf{u}_n = \mathbf{V}^T(\mathbf{x}_n - \hat{\boldsymbol{\mu}}).$$

Collecting latent variables in matrix $\mathbf{U} = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_N \end{pmatrix}^T$ and by denoting $\tilde{\mathbf{x}}_n = \mathbf{x}_n - \hat{\boldsymbol{\mu}}$ the projection can be written as $\mathbf{U} = \widetilde{\mathbf{X}}\mathbf{V}$. For $K = D$, $\mathbf{V}\mathbf{V}^T = \mathbf{I}$ and $\widetilde{\mathbf{X}} = \mathbf{U}\mathbf{V}^T$. For $K < D$ the equality does not hold, assuming that the rank of $\mathbf{X}$ is $D$, but the approximation $\widetilde{\mathbf{X}} \approx \mathbf{U}\mathbf{V}^T$ is still optimal in some sense. Specifically Pearson (1901) proves, that the PCA solution is the one that minimizes the cost function $J$ written as

$$J = \sum_{n=1}^{N} ||\mathbf{x}_n - \hat{\mathbf{x}}_n||^2 = \sum_{n=1}^{N} ||\mathbf{x}_n - \mathbf{V}\mathbf{u}_n - \boldsymbol{\mu}||^2, \tag{12}$$

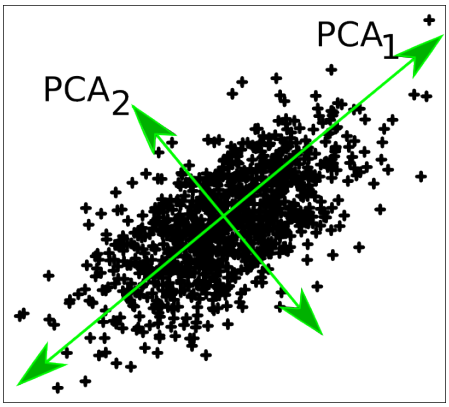



Figure 2: Illustration of PCA. The two PCA projections are plotted for the simulated 2-dimensional data. The first projection captures maximal variance while the second is constrained to be orthogonal to the first. The lengths of the arrows are scaled according to the captured variance.

where the approximation $\hat{\mathbf{x}}$ of the original data $\mathbf{x}$ is constrained to be low-rank.

In order to proceed towards probabilistic modeling, it is next shown how the cost function $J$ of (12) stems from assuming a probabilistic model for $\mathbf{x}$.

### 3.2.1 Maximum likelihood principal component analysis

If the matrix elements are assumed to be conditionally independent and exchangeable given the parameters , the probabilistic model for the data can be written as

$$p(\mathbf{x}_1, \ldots . \mathbf{x}_N | \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) = p(\mathbf{X}|\boldsymbol{\Theta}) = \prod_{n=1}^{N} p(\mathbf{x}_n|\boldsymbol{\theta}_n) = \prod_{n=1}^{N}\prod_{d=1}^{D} p(x_{nd}|\theta_{nd}). \tag{13}$$

The assumption of conditional independence can be justified if $\boldsymbol{\Theta}$ is flexible enough to capture the dependencies in $\mathbf{x}$. The PCA model is obtained by constraining $\boldsymbol{\Theta} = \begin{pmatrix} \boldsymbol{\theta}_1 & \boldsymbol{\theta}_2 & \ldots & \boldsymbol{\theta}_N \end{pmatrix}^T$, where $\boldsymbol{\theta}_n = \mathbf{V}\mathbf{u}_n + \boldsymbol{\mu}$, and assuming that $p(x|\theta)$ corresponds to the Gaussian distribution with known variance.

Following ML inference, the complete data log-likelihood is written as

$$\begin{aligned}
\mathcal{L} &= \ln p(\mathbf{X}|\boldsymbol{\Theta}) = \sum_{n=1}^{N} \ln p(\mathbf{x}_n|\boldsymbol{\theta}_n) = \sum_{n=1}^{N}\sum_{d=1}^{D} \ln p(x_{nd}|\theta_{nd}) \\
&= \sum_{n=1}^{N}\sum_{d=1}^{D} -\ln\sqrt{2\pi\sigma^2} - \frac{1}{2\sigma^2}(x_{nd} - \theta_{nd})^2 \\
&= -ND\ln\sqrt{2\pi\sigma^2} - \frac{1}{2\sigma^2}\sum_{n=1}^{N} ||\mathbf{x}_n - \mathbf{V}\mathbf{u}_n - \boldsymbol{\mu}||^2.
\end{aligned}$$

It can be shown that the log-likelihood corresponds to the cost function of the PCA model, $\mathcal{L} = -J$, omitting terms that do not depend on the parameters and assuming $\sigma^2 = 1/2$.

## 3.3 Exponential family principal component analysis

The immediate extension of PCA to the exponential family is given by noting that the Gaussian distribution belongs to the exponential family. The generalization of PCA to the exponential family (EPCA) retains from PCA the property that the parameters of the distribution are represented as $\boldsymbol{\theta} = \mathbf{V}\mathbf{u} + \boldsymbol{\mu}$, while the likelihood function $p(x_{nd}|\theta_{nd})$ changes according to different assumptions on the noise model (Section 3.1). EPCA provides a unified framework for PCA for different data types. The only difference in the likelihood function between different distributions in exponential family is the function $g(\cdot)$, because $h(\cdot)$ does not depend on parameters.

An important difference between different assumptions for $p(\mathbf{x}|\boldsymbol{\theta})$ is in predictions. Wrong assumptions for $p(\mathbf{x}|\boldsymbol{\theta})$ can lead to predictions out of the domain of data. For example, in missing value imputation task for binary data the Gaussian assumption leads to predictions in $\mathbb{R}$, while all the values should be exactly 0 or 1. If data is known to be binary it is recommended to make the assumption that the likelihood corresponds to the Bernoulli distribution, because then the only kind of noise possible is bit flips, i.e., 1 changes to 0 or vice versa.

### 3.3.1 Maximum likelihood inference for exponential family principal component analysis

To find the maximum likelihood estimates for $\mathbf{U}$ and $\mathbf{V}$ for an observed data matrix $\mathbf{X}$, the data log likelihood needs to be maximized. To simplify the notation denote $\mathbf{U} := \begin{pmatrix} \mathbf{U} & \mathbf{1} \end{pmatrix}$ and $\mathbf{V} := \begin{pmatrix} \mathbf{V} & \boldsymbol{\mu} \end{pmatrix}$, incorporating the mean parameter in $\mathbf{U}\mathbf{V}^T$. Maximization of the log likelihood, written in vector-matrix form,

$$\mathcal{L} = Tr[\mathbf{X}(\mathbf{U}\mathbf{V}^T)^T] - \sum_{nd} g(\mathbf{U}\mathbf{V}^T) \tag{14}$$

can be performed by finding a point satisfying $\nabla\mathcal{L} = \mathbf{0}$. Above $Tr[\mathbf{C}] = \sum_i \mathbf{C}_{ii}$ is used to denote the trace of the square matrix $\mathbf{C}$. The notation for $g(\mathbf{C})$ (also for $g'(\mathbf{C})$) is overloaded, for a matrix argument it corresponds to element-wise application. Local optima of (14) are defined as points that satisfy the condition, $\nabla\mathcal{L}^l = \mathbf{0}$. However, there may exist a better solution corresponding to the global maximum $\mathcal{L}^*$ with the property $\mathcal{L}^* \geq \mathcal{L}^l \ \forall l$.

The gradient of (14) with respect to the latent variables can be written as

$$\nabla_{\mathbf{U}}\mathcal{L} = \big(\mathbf{X} - g'(\mathbf{U}\mathbf{V}^T)\big)\mathbf{V},$$

and with respect to the projection matrix as

$$\nabla_{\mathbf{V}}\mathcal{L} = \mathbf{U}^T\big(\mathbf{X} - g'(\mathbf{U}\mathbf{V}^T)\big).$$

Maximum likelihood inference for EPCA thus results in matrix factorization written as $\mathbf{X} \approx g'(\mathbf{U}\mathbf{V}^T)$. For suitably large $K$ the approximation becomes the equivalence $\mathbf{X} = g'(\mathbf{U}\mathbf{V}^T)$. The derivative of the log cumulant function $g(\cdot)$ hence provides a link between the data and the natural parameter space. For classical PCA with

Gaussian likelihood the parameter space and the data space are equivalent (Section 3.1) and hence $\mathbf{X} \approx \mathbf{UV}$. That is, the parameters and data have linear relationship and inference can be carried out by solving linear systems (Section 3.2). In the general case the relationship between the parameters and the data is not linear and iterative optimization methods need to be applied (discussed in Section 6.1).

## 3.4 Probabilistic principal component analysis

Despite the use of $p(\mathbf{x}|\boldsymbol{\theta})$ to define the PCA cost function, it does not provide full probabilistic model as the variance term was assumed known. Next it is shown how PCA can be interpreted as a a full probabilistic model. Even though the presented details apply only for the Gaussian model, EPCA has essentially the same properties. This section gives more detailed view of the PCA model, essentially, answering questions like what kind of dependencies PCA can capture and to what kind of data it is suitable to apply for.

In Section 3.1 only univariate exponential family distributions were presented in detail. Details for the multivariate distributions such as multinomial, Gaussian with unknown variance parameter, and multivariate Gaussian, that do belong to the exponential family, were omitted but can be found from (Bernardo and Smith, 2000). For now on $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is used to denote the multivariate Gaussian density with mean parameter $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

The PCA model can be written as

$$\mathbf{x} = \mathbf{Vu} + \boldsymbol{\mu} + \boldsymbol{\epsilon}, \tag{15}$$

where $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$ is a noise term, and the subscripts used for different data points are removed for clarity. Essentially, $\boldsymbol{\theta} = \mathbf{Vu} + \boldsymbol{\mu}$ is assumed to be a noiseless version of the observed noisy data point $\mathbf{x}$.

By assuming that the latent points follow the prior distribution $p(\mathbf{u}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, one can write the mean of $\mathbf{x}$ under the model assumptions with known $\mathbf{V}$ and $\boldsymbol{\mu}$ as

$$\mathbb{E}[\mathbf{x}] = \mathbb{E}[\mathbf{Vu} + \boldsymbol{\mu} + \boldsymbol{\epsilon}] = \boldsymbol{\mu}, \tag{16}$$

because $\mathbb{E}[\mathbf{u}] = \mathbb{E}[\boldsymbol{\epsilon}] = \mathbf{0}$, and because expectation is a linear operator. The covariance of $\mathbf{x}$ can be written as

$$\begin{aligned}
\mathbb{E}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^T] &= \mathbb{E}[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T] && (17)\\
&= \mathbb{E}[(\mathbf{Vu} + \boldsymbol{\epsilon})(\mathbf{Vu} + \boldsymbol{\epsilon})^T] && (18)\\
&= \mathbb{E}[\mathbf{Vuu}^T\mathbf{V}^T + \mathbf{Vu}\boldsymbol{\epsilon}^T + \boldsymbol{\epsilon}\mathbf{u}^T\mathbf{V}^T + \boldsymbol{\epsilon}\boldsymbol{\epsilon}^T] && (19)\\
&= \mathbf{VV}^T + \sigma^2\mathbf{I}, && (20)
\end{aligned}$$

because the noise and the latent variables are assumed to be independent, $\mathbb{E}[\mathbf{u}\boldsymbol{\epsilon}^T] = \mathbb{E}[\boldsymbol{\epsilon}\mathbf{u}^T] = \mathbf{0}$, $\mathbb{E}[\mathbf{uu}^T] = \mathbf{I}$, and $\mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T] = \sigma^2\mathbf{I}$. It can be seen how $\mathbf{V}$ captures the covariances of $\mathbf{x}$. By the assumption of Gaussian noise and latent variables it can be shown (Bishop, 2006) that the marginal likelihood, or equivalently the data distribution, is another Gaussian distribution written as

$$\mathbf{x}|\mathbf{V}, \boldsymbol{\mu}, \sigma^2 \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{VV}^T + \sigma^2\mathbf{I}). \tag{21}$$

This is directly obtained by solving

$$p(\mathbf{x}|\mathbf{V}, \boldsymbol{\mu}, \sigma^2) = \int p(\mathbf{x}|\mathbf{u}, \mathbf{V}, \boldsymbol{\mu}, \sigma^2)p(\mathbf{u})d\mathbf{u}. \tag{22}$$

Tipping and Bishop (1999) show that maximizing the marginal likelihood with respect to the parameters, $\boldsymbol{\mu}$, $\mathbf{V}$ and $\sigma^2$, leads to the PCA solution. The maximum likelihood estimates are

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n \tag{23}$$

$$\hat{\mathbf{V}} = \mathbf{W}_K(\boldsymbol{\Lambda}_K - \hat{\sigma}^2\mathbf{I})^{1/2}\mathbf{R} \tag{24}$$

$$\hat{\sigma}^2 = \frac{1}{D-K}\sum_{n=K+1}^{D}\lambda_n, \tag{25}$$

where $\mathbf{W}_K$ corresponds to $K$ leading eigenvectors of $\mathbf{C}$ and $\mathbf{R}$ is an arbitrary $K \times K$ orthogonal matrix. The parametrization of the projection matrix is defined up to a rotation of the PCA solution, as can be seen by writing

$$\hat{\mathbf{V}}\hat{\mathbf{V}}^T = \mathbf{V}\mathbf{R}\mathbf{R}^T\mathbf{V}^T = \mathbf{V}\mathbf{V}^T,$$

since $\mathbf{R}\mathbf{R}^T = \mathbf{I}$. However, the actual PCA projections can be solved up to a sign change by replacing the sample covariance matrix with $\hat{\mathbf{V}}\hat{\mathbf{V}}^T + \hat{\sigma}^2\mathbf{I}$ for the PCA algorithm in Section 3.2.

The model is named probabilistic PCA (PPCA) and it is called generative because new data can be generated from it. This is important property, and illustrations of it are given in Section 3.1.

Tipping and Bishop (1999) further show that the predictive distribution is yet another Gaussian distribution,

$$p(\mathbf{u}^*|\mathbf{x}^*) = N(\mathbf{u}^*|\mathbf{M}^{-1}\mathbf{V}^T(\mathbf{x}^* - \boldsymbol{\mu}), \sigma^2\mathbf{M}),$$

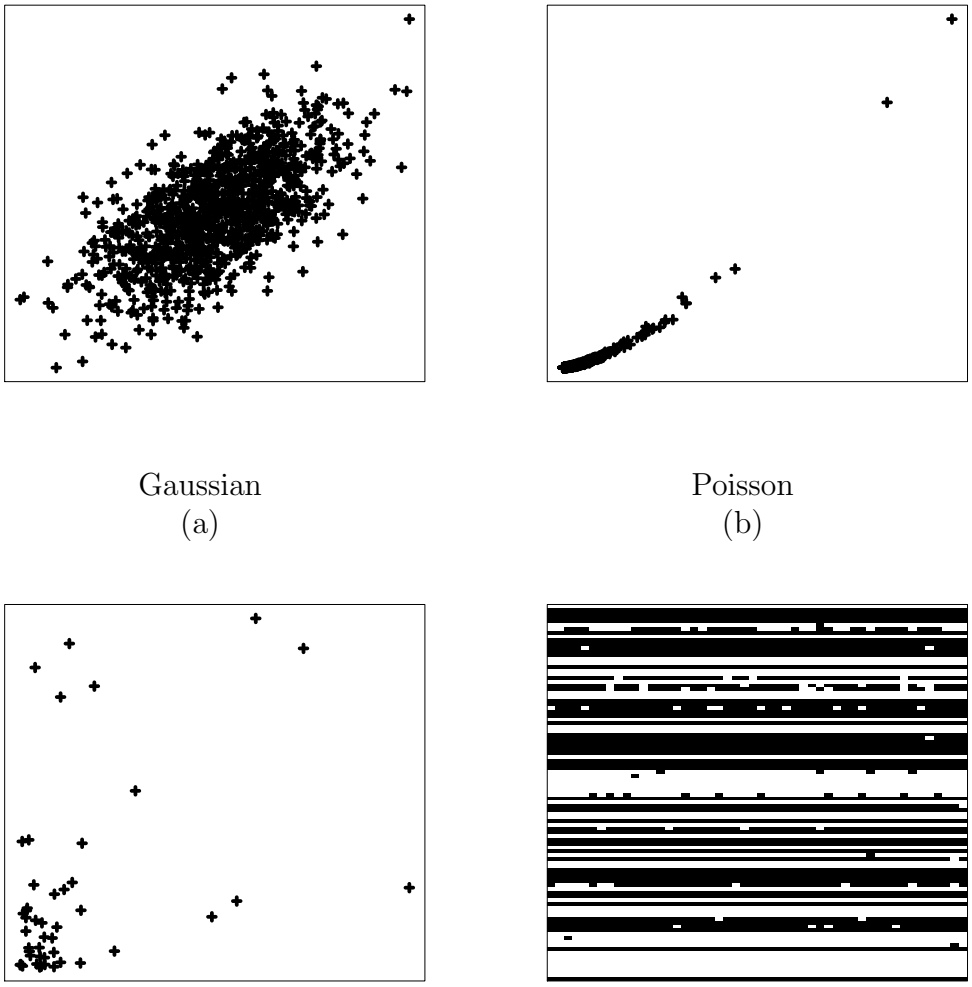
where $\mathbf{M} = \mathbf{V}^T\mathbf{V} + \sigma^2\mathbf{I}$. The mean of the distribution is equivalent to the PCA subspace, up to rotation $\mathbf{R}$.

Comparing PPCA to PCA, the most significant difference is the global noise term, $\sigma^2$. Further, it was shown above that traditional PCA makes tacit assumption of Gaussian data and latent variables.

## 3.5 Demonstrations

For the PPCA model exact marginalization of the latent variables can be done, leading to the multivariate Gaussian distribution (21). However, for the other members of the exponential family this marginalization cannot be performed in closed form.

For demonstration, different data sets are drawn from the EPCA model for various data types, and visualized in Figure 3 retaining the assumption that $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. From Figure 3 it can be seen how the distributional assumptions completely change the shape of the data.



Gaussian
(a)

Poisson
(b)

Exponential
(c)

Bernoulli
(d)

Figure 3: Data generated from the EPCA model with $K = 1$. Gaussian noise corresponds to (a), (b) corresponds to Poisson and (c) to exponentially distributed noise. In figure (d) there is a 50-dimensional binary data matrix plotted to represent the dependencies because two-dimensional binary data has only 4 different combinations and would not be interesting. It can be seen that the generated data is very different for the different members of the exponential family. For the Poisson distribution we see examples of rare events and exponentially distributed variables often obtain large values.

## 3.6 Related methods

The likelihood $p(\mathbf{x}|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is low-rank, is used in many other famous machine learning methods for matrix factorization. The methods differ in multiple ways: in the form of the likelihood function as shown above, the factorization of $\boldsymbol{\theta}$, or the constraints and prior distributions set for the low-rank decompositions.

In this thesis, linear factorization of $\boldsymbol{\theta}$ is considered. However, non-linearities can be taken into account as well. The problem is then how to define these non-linearities. Lawrence (2005) considers extension of PCA to Gaussian processes. Instead of marginalizing the latent variables, marginalization of $\mathbf{V}$ is performed assuming a column-wise Gaussian prior. Further, the kernel trick is used to make the model non-linear. Another non-linear Bayesian approach is given by Lian (2009); non-linear dimensionality reduction is achieved by applying in the latent space local projections that are smoothed with a Markov Random Field-type prior.

For EPCA the relationship between the parameters and the data is expressed with a non-linear function. Using suitable non-linearities while still assuming Gaussian data can be seen as heuristic approach for taking correct noise type into account (see for example (Salakhutdinov and Mnih, 2008; Ma et al., 2008)). Suitable non-linearities are readily proposed by $g'(\cdot)$. Mathematically this can be written as the following observation equation

$$\mathbf{x} = f(\mathbf{V}\mathbf{u}) + \boldsymbol{\epsilon},$$

where $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$ and $f(\cdot)$ is a non-linear function. Incorporation of non-linearities to the squared loss function complicates inference by introducing local minima (Gordon, 2002).

By changing the prior distribution of the latent variables to Dirichlet or multinomial the task of dimensionality reduction is turned into clustering, for example (Heller et al., 2008). Bingham et al. (2009) compare two different priors for the latent variables, continuous and constrained, concluding that by constraining the latent variables they become competitive and in the other case they work in collaboration. This affects the interpretation of components. In non-negative matrix factorization the components of $\boldsymbol{\theta}$ are constrained to be positive (Lee and Seung, 2001).

In general, determining the noise distribution can be challenging. Guo and Schuurmans (2008) proposed a more flexible framework where the function $g(\cdot)$ is solved by the sample-based approximation

$$g(\theta) \approx \ln\left(\frac{1}{N}\sum_{i=1}^{N}\exp(x_i\theta)\right).$$

The observed data hence determines the distribution in question. Guo (2009) also incorporates this approximation. Another approach would be to use some model selection procedure to choose the most likely distribution for the data.

# 4 Models for paired data

The methods considered so far are suitable for the analysis of a single data matrix. One of the main contributions of this thesis is to present how multi-source learning methods arise as special cases of the EPCA model.

Two random variables, $\mathbf{y}_1 \in \mathbb{K}^{D_1}$ and $\mathbf{y}_2 \in \mathbb{K}^{D_2}$, are paired if the samples in the two co-occur. Co-occurring samples are generated in pairs of items, $\mathbf{y}_1$ and $\mathbf{y}_2$. By concatenating the two sources as $\mathbf{x}^T = \begin{pmatrix} \mathbf{y}_1^T & \mathbf{y}_2^T \end{pmatrix}$ several projection methods for paired data sources can be written as EPCA of $\mathbf{x}$, that is, as factorizations of the form $\boldsymbol{\theta} = \mathbf{V}\mathbf{u}$.

Different kinds of models are obtained by specifying different constraints on $\mathbf{V}$. Many of the decisions in practical modeling, such as the choice of prior distributions and inference algorithm, are independent of such restrictions imposed on $\mathbf{V}$, and hence the unified framework helps in developing practical algorithms for various paired data analysis tools.

Below only the likelihood functions for the proposed models are presented, while the corresponding prior distributions are presented in Section 5, as these are shared between all the methods.

## 4.1 Supervised exponential family principal component analysis

Supervised PCA (SPCA) is the simplest model for paired data. One of the sources, say $\mathbf{y}_1$, is treated as a target variable, and the task is to find a low-dimensional representation of $\mathbf{y}_2$ that helps in predicting the target.

The task is termed as supervised dimensionality reduction. Instead of finding a latent variable description of $\mathbf{y}_2$, the lower dimensional manifold is obtained for prediction of $\mathbf{y}_1$. EPCA as preprocessing for $\mathbf{y}_2$ alone would not take the target information into account, and hence would not necessarily produce latent variables predictive of $\mathbf{y}_1$.

The original SPCA formulation (Yu et al., 2006) as well as the supervised EPCA (Guo, 2009; SEPCA) follow this idea, the crucial difference being that the latter makes the correct distribution assumption for the target variables. If $\mathbf{y}_1$ with $D_1 = 1$ is continuous the task is called regression and if binary the task is classification. For the regression a typical assumption for $\mathbf{y}_1$ is a Gaussian distribution while for binary classification Bernoulli distribution is a more justified choice. For $D_1 > 1$ multi-regression or multi-classification tasks arise that can be defined as special cases of multi-task learning (Caruana, 1997): multiple prediction tasks share the same input, the latent variables.

Due to the assumption of conditional independence between $\mathbf{y}_1$ and $\mathbf{y}_2$, given the parameters, the SEPCA likelihood can be written as

$$p(\mathbf{x}|\mathbf{u}, \mathbf{V}, \boldsymbol{\mu}) = p(\mathbf{y}_1|\mathbf{u}, \mathbf{V}_1, \boldsymbol{\mu}_1)p(\mathbf{y}_2|\mathbf{u}, \mathbf{V}_2, \boldsymbol{\mu}_2)$$

or, more clearly, to clarify the dependence between data and parameters, as

$$p(\mathbf{x}|\boldsymbol{\theta}) = p(\mathbf{y}_1|\boldsymbol{\theta}_1)p(\mathbf{y}_2|\boldsymbol{\theta}_2).$$

where $\boldsymbol{\theta} = \begin{pmatrix} \boldsymbol{\theta}_1 \\ \boldsymbol{\theta}_2 \end{pmatrix} = \mathbf{V}\mathbf{u} + \boldsymbol{\mu}$,

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}$$

and

$$\mathbf{V} = \begin{pmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{pmatrix}$$

so that the columns are split according to the features in $\mathbf{x}$.

Briefly put, SEPCA is EPCA of $\mathbf{x}$. The features are treated equally and hence this approach provides weak supervision as the model aims to capture all dependencies between the elements of $\mathbf{x}$. The predictive performance of such a model improves if one does not attempt to model $\mathbf{y}_2$ perfectly; after all, the ultimate task is to predict $\mathbf{y}_1$ and the covariates should be modeled only to the degree they help in that task. Rish et al. (2008) proposed an approach to weight the generative parts, resulting in the model likelihood

$$p(\mathbf{x}|\boldsymbol{\theta}) = p(\mathbf{y}_1|\boldsymbol{\theta}_1)p(\mathbf{y}_2|\boldsymbol{\theta}_2)^{\alpha}, \tag{26}$$

where $\alpha$ controls the relative importance of modeling the two sources. When small values are chosen for $\alpha$ less modeling power is spent on the covariates, resulting in increased predictive performance.

Instead of treating $\alpha$ as an arbitrary control parameter, it can be interpreted as a fixed variance parameter in the general exponential family formulation (Gelman et al., 2004). Dropping the assumption of identical variance for all features, (7) can be written as

$$p(\mathbf{x}|\boldsymbol{\theta}) = \prod_{d=1}^{D} \exp\Big(w_i\big(x_i\theta_i + \ln h(x_i) - g(\theta_i)\big)\Big),$$

where $\mathbf{w}$ is a vector consisting of ones for $\mathbf{y}_1$ and $\alpha$'s for $\mathbf{y}_2$,

$$\mathbf{w}^T = \begin{pmatrix} 1 & \dots & 1 & \alpha & \dots & \alpha \end{pmatrix}.$$

The interpretation has close relationship to maximum likelihood factor analysis (FA) (Bishop, 2006). In FA the noise follows $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is diagonal matrix with separate elements on its diagonal, while PCA assumes that all elements are equal. For Gaussian data the $\mathbf{w}$ parameter can be recognized as an inverse variance parameter. However, even this formulation does not provide easy ways of inferring $\alpha$ from data, since changing it would affect the normalization constant of the distribution $p(\mathbf{x}|\boldsymbol{\theta})$.

### 4.1.1 Applications of supervised exponential family principal component analysis

In general SEPCA can be used in data integration, combining multiple sources of information to improve prediction accuracy. Williamson and Ghahramani (2008) and Ma et al. (2008) considered joint models for data combination in recommender

systems assuming Gaussian data, while Singh and Gordon (2008) present more general framework performing modeling in the exponential family.

Recommender systems aim to suggest for users movies they would like to see based on the movie ratings of other users. This is a missing value imputation problem. Incorporating auxiliary data of users and/or movies may improve prediction accuracy. Ma et al. (2008) considered fusion of social network data of users to improve movie recommendation accuracy. Motivation is that friends usually have similar taste and they recommend movies to each others.

SEPCA can also be seen as incorporating 'background knowledge' to EPCA matrix factorization. Typically the latent variables are assumed some simple prior distribution $p(\mathbf{u})$. Recent approach of Bo and Schmisesku (2009) introduce so-called supervised latent variables, that is, the latent variables depend on $\mathbf{y}_2$. Mathematically the assumption can be written as $p(\mathbf{u}|\mathbf{y}_2)$. However, the two approaches are equivalent. This can be seen by writing

$$p(\mathbf{y}_1|\mathbf{u})p(\mathbf{y}_2|\mathbf{u})p(\mathbf{u}) = p(\mathbf{y}_1|\mathbf{u})\frac{p(\mathbf{u}|\mathbf{y}_2)p(\mathbf{y}_2)}{p(\mathbf{u})}p(\mathbf{u}) = p(\mathbf{y}_1|\mathbf{u})p(\mathbf{u}|\mathbf{y}_2)p(\mathbf{y}_2).$$

## 4.2 Exponential family partial least squares

An alternative way of improving the predictive performance in supervised learning tasks is to allow the covariates to have structured noise that is independent of the target variable. This leads naturally to a classical linear supervised dimensionality reduction method of partial least squares (PLS) and its probabilistic variants (Gustafsson, 2001; Nounou et al., 2002). These models are restricted to Gaussian data. In this thesis, the correct data type is taken into account, introducing the exponential family partial least squares (EPLS).

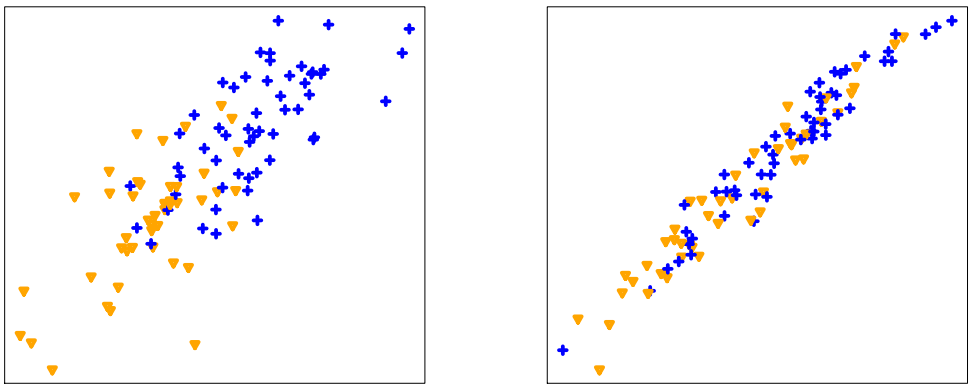
The key idea in PLS is that not all variation in $\mathbf{y}_2$ is relevant for predicting $\mathbf{y}_1$. A novel way incorporating that knowledge is proposed in the model by restricting some of the components to only model $\mathbf{y}_2$. By factoring $\mathbf{u}^T = \begin{pmatrix} \mathbf{u}_S^T & \mathbf{u}_2^T \end{pmatrix}$ and $\mathbf{V}$ as

$$\mathbf{V} = \begin{pmatrix} \mathbf{V}_{S1} & \mathbf{0} \\ \mathbf{V}_{S2} & \mathbf{V}_2 \end{pmatrix},$$

where $S$ indicates variables shared between the data sources, the model can still be written as $\boldsymbol{\theta} = \mathbf{V}\mathbf{u} + \boldsymbol{\mu}$. The model complexity is governed by fixing the ranks of the various parts. Denoting the rank of $\mathbf{u}_S$ by $K_S$ and the rank of $\mathbf{u}_2$ by $K_2$, the zeros in $\mathbf{V}$ make sure the last $K_2$ columns of $\mathbf{u}$ will have no effect on $\mathbf{y}_1$. In more intuitive terms, the parameters can equivalently be written as

$$\begin{aligned} \boldsymbol{\theta}_1 &= \mathbf{V}_{S1}\mathbf{u}_S + \boldsymbol{\mu}_1 \\ \boldsymbol{\theta}_2 &= \mathbf{V}_{S2}\mathbf{u}_S + \mathbf{V}_2\mathbf{u}_2, + \boldsymbol{\mu}_2. \end{aligned} \tag{27}$$

which makes explicit the assumption that all variation in the target variable must come from the shared latent sources, while the covariates are created as an additive sum of the shared and source-specific variation.



(a) (b)

Figure 4: In (a) the Gaussian covariate data with $D_2 = 2$ and the binary target data with $D_1 = 1$ is generated from the SEPCA model with $D_1 = 1$ and $K = 1$. Different symbols and colors indicate binary $\mathbf{y}_1$. Clear structure can be seen from the covariate data; the classes are separated. In (b) the data is created from the EPLS model by adding structured noise to $\mathbf{y}_2$ ($K_2 = 1$). Due to the addition of the noise, the class structure is not anymore visible, while (b) contains the same information as (a). The example demonstrates the need for more advanced exploratory data analysis methods. In the experiments (Section 7.1) it is demonstrated how EPLS can find the correct shared subspace, while SEPCA fails.

In the experiments (Section 7.1) it will be shown how this modeling assumption reduces the number of shared components needed for predicting $\mathbf{y}_1$ better than the exponent $\alpha$ in SEPCA. This improves the interpretability of the results. Besides making good predictions, the actual projections can be used to infer what aspects of $\mathbf{y}_2$ are predictive of the target variable.

In figure 4 the difference between SEPCA and EPLS models is demonstrated by generating data from the models. See the caption for more detailed explanation.

### 4.2.1 Special case for Gaussian data

If Gaussian data is assumed and if the latent variables follow $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, marginalization of the latent variables, as in (22), results in

$$\mathbf{x}|\mathbf{V}, \boldsymbol{\mu} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{V}\mathbf{V}^T + \sigma^2\mathbf{I}),$$

where

$$\mathbf{V}\mathbf{V}^T = \begin{pmatrix} \mathbf{V}_{S1}\mathbf{V}_{S1}^T & \mathbf{V}_{S1}\mathbf{V}_{S2}^T \\ \mathbf{V}_{S2}\mathbf{V}_{S1}^T & \mathbf{V}_{S2}\mathbf{V}_{S2}^T + \mathbf{V}_2\mathbf{V}_2^T \end{pmatrix}. \tag{28}$$

Now it can easily be seen how the dependencies between $\mathbf{y}_1$ and $\mathbf{y}_2$ are modeled only with the shared components. From the structure of the covariance matrix it

can be deduced that $K_2$ should be set high enough in order to prevent the shared components to capture covariate-specific variation. The structure of covariance matrix for the PCA model is identical to (28), except for omitting $\mathbf{V}_2\mathbf{V}_2^T$. This means that PCA captures all dependencies, ignoring whether they are shared or not.

For distributions other than Gaussian, marginalization of the latent variables can no longer be performed in closed form, but the presented model properties remain: the dependencies between $\mathbf{y}_1$ and $\mathbf{y}_2$ are still captured in $\mathbf{V}_{S1}$ and $\mathbf{V}_{S2}$. The covariate-specific variables can be seen as nuisance parameters that usually are not of interest but necessary to extract the correct components.

## 4.3 Exponential family data fusion

Going beyond mere prediction problems, a common task in analysis of paired data is finding what is shared between the two data sources. This is a kind of data fusion task: compress two data sources into a representation that captures the commonalities between the two. Alternatively, one can further represent the source-specific variation present in each of the sources separately, independent of the other source. The problem is traditionally solved by canonical correlation analysis (Hotelling, 1936), or its kernelized variant (Bach and Jordan, 2002), that have been applied to a range of practical problems such as extracting shared semantics of document translations (Vinokourov et al., 2003) and discovering dependencies between images and associated text to be used as preprocessing for classification (Farquhar et al., 2006).

PCA seeks projections that capture maximal variance in the projected space, whereas CCA can be seen as a method seeking for two projections maximizing correlation between the projected data. Similarly to PPCA presented in Section 2, it can be shown that probabilistic CCA assumes Gaussian latent variables and is equivalent to assuming a certain Gaussian model for the data (Bach and Jordan, 2005). In this thesis this assumption is removed presenting a novel generalization of CCA to the exponential family, termed ECCA for brevity. This is necessary, for example, for text analysis with the generative approach, since text documents are naturally described as binary collections of word occurrences or as count data.

First the CCA model and the corresponding probabilistic interpretation are explained. Secondly ECCA is presented. Finally the combination of EPLS and ECCA models is presented that result in supervised ECCA (SECCA). Both ECCA and SECCA are novel contributions of this thesis.

### 4.3.1 Canonical correlation analysis

CCA aims to find linear transformations for the two random variables, $\mathbf{y}_1$ and $\mathbf{y}_2$, with $N$ realizations collected in matrices $\mathbf{Y}_1$ and $\mathbf{Y}_2$ such that the projected data, $\tilde{\mathbf{y}}_1 = \mathbf{Y}_1\mathbf{w}_1$ and $\tilde{\mathbf{y}}_2 = \mathbf{Y}_2\mathbf{w}_2$, is maximally correlated. We denote the sample covariance matrix of $\mathbf{X} = \begin{pmatrix} \mathbf{Y}_1 & \mathbf{Y}_2 \end{pmatrix}$ as

$$\mathbf{C} = \frac{1}{N}\begin{pmatrix} \mathbf{Y}_1^T\mathbf{Y}_1 & \mathbf{Y}_1^T\mathbf{Y}_2 \\ \mathbf{Y}_2^T\mathbf{Y}_1 & \mathbf{Y}_2^T\mathbf{Y}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22} \end{pmatrix}$$

assuming that the data sets are centered. Correlation for the projected data can be written as

$$\begin{aligned}\rho =& \frac{\tilde{\mathbf{y}}_1^T \tilde{\mathbf{y}}_2}{\sqrt{\tilde{\mathbf{y}}_1^T \tilde{\mathbf{y}}_1}\sqrt{\tilde{\mathbf{y}}_2^T \tilde{\mathbf{y}}_2^T}} \\ =& \frac{\mathbf{w}_1^T \mathbf{Y}_1^T \mathbf{Y}_2 \mathbf{w}_2}{\sqrt{\mathbf{w}_1^T \mathbf{Y}_1^T \mathbf{Y}_1 \mathbf{w}_1}\sqrt{\mathbf{w}_2^T \mathbf{Y}_2^T \mathbf{Y}_2 \mathbf{w}_2}} \\ =& \frac{N\mathbf{w}_1^T \mathbf{\Sigma}_{12} \mathbf{w}_2}{\sqrt{N\mathbf{w}_1^T \mathbf{\Sigma}_{11} \mathbf{w}_1}\sqrt{N\mathbf{w}_2^T \mathbf{\Sigma}_{22} \mathbf{w}_2}} \\ =& \frac{\mathbf{w}_1^T \mathbf{\Sigma}_{12} \mathbf{w}_2}{\sqrt{\mathbf{w}_1^T \mathbf{\Sigma}_{11} \mathbf{w}_1}\sqrt{\mathbf{w}_2^T \mathbf{\Sigma}_{22} \mathbf{w}_2}}. \end{aligned} \tag{29}$$

Maximization of (29) can be done by constraining $\mathbf{w}_1^T \mathbf{\Sigma}_{11} \mathbf{w}_1^T = \mathbf{w}_2^T \mathbf{\Sigma}_{22} \mathbf{w}_2^T = 1$ and applying the technique of constrained optimization. Multiple projections $\mathbf{w}_{1i}$ and $\mathbf{w}_{2i}$ with $i \leq \min(D_1, D_2)$ are constrained to be uncorrelated, $\mathbf{w}_{1i}^T \mathbf{\Sigma}_{11} \mathbf{w}_{1j}^T = \mathbf{I}_{ij}$, and similarly for $\mathbf{w}_{2i}$. The solution for $K$ vectors, collected as columns in $\mathbf{W}_1$ and $\mathbf{W}_2$, corresponds to the $K$ first leading eigenvectors of the generalized eigenvalue problem

$$\begin{pmatrix} \mathbf{0} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{pmatrix} = \rho \begin{pmatrix} \mathbf{\Sigma}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{pmatrix}.$$

Proof and details of the constrained optimization are omitted, but can be found from (Shawe-Taylor and Cristianini, 2004).

### 4.3.2 Probabilistic canonical correlation analysis

CCA can be interpreted as a maximum likelihood solution of a certain probabilistic model. Latent variables are defined as $\mathbf{u}^T = \begin{pmatrix} \mathbf{u}_S^T & \mathbf{u}_1^T & \mathbf{u}_2^T \end{pmatrix}$ and

$$\mathbf{V} = \begin{pmatrix} \mathbf{V}_{S1} & \mathbf{V}_1 & \mathbf{0} \\ \mathbf{V}_{S2} & \mathbf{0} & \mathbf{V}_2 \end{pmatrix}$$

for the model

$$\mathbf{x}|\mathbf{u}, \mathbf{V}, \boldsymbol{\mu} \sim \mathcal{N}(\mathbf{V}\mathbf{u} + \boldsymbol{\mu}, \sigma^2 \mathbf{I}) \tag{30}$$

following the presentation of Archambeau and Bach (2009).

Assuming that latent variables follow $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, marginalization of the latent variables results in

$$\mathbf{x}|\mathbf{V}, \boldsymbol{\mu} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{V}\mathbf{V}^T + \sigma^2 \mathbf{I}), \tag{31}$$

where

$$\mathbf{V}\mathbf{V}^T = \begin{pmatrix} \mathbf{V}_{S1}\mathbf{V}_{S1}^T + \mathbf{V}_1\mathbf{V}_1^T & \mathbf{V}_{S1}\mathbf{V}_{S2}^T \\ \mathbf{V}_{S2}\mathbf{V}_{S1}^T & \mathbf{V}_{S2}\mathbf{V}_{S2}^T + \mathbf{V}_2\mathbf{V}_2^T \end{pmatrix}.$$

For the above model Bach and Jordan (2005), denoting $\mathbf{\Psi}_1 = \mathbf{V}_1\mathbf{V}_1^T + \sigma^2\mathbf{I}$ and $\mathbf{\Psi}_2 = \mathbf{V}_2\mathbf{V}_2^T + \sigma^2\mathbf{I}$, show the connection to standard CCA. More precisely, maximum likelihood estimate of (31) corresponds to the projections found by traditional CCA, up to rotation and scaling.

According to Theorem 2 by Bach and Jordan (2005), the maximum likelihood estimates for the parameters of the model in (31) are

$$\begin{aligned}
\widehat{\mathbf{V}_{S1}} &= \mathbf{\Sigma}_{11}\mathbf{W}_1\mathbf{M}_1 \\
\widehat{\mathbf{V}_{S2}} &= \mathbf{\Sigma}_{22}\mathbf{W}_2\mathbf{M}_2 \\
\widehat{\mathbf{\Psi}}_1 &= \mathbf{\Sigma}_{11} - \widehat{\mathbf{V}_{S1}}\widehat{\mathbf{V}_{S1}}^T \\
\widehat{\mathbf{\Psi}}_2 &= \mathbf{\Sigma}_{22} - \widehat{\mathbf{V}_{S2}}\widehat{\mathbf{V}_{S2}}^T \\
\hat{\boldsymbol{\mu}}_1 &= \boldsymbol{\mu}_1 \\
\hat{\boldsymbol{\mu}}_2 &= \boldsymbol{\mu}_2,
\end{aligned}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are the CCA projection matrices, $\mathbf{M}_1$ and $\mathbf{M}_2$ are arbitrary matrices such that $\mathbf{M}_1\mathbf{M}_2 = \mathbf{P}$ and the spectral norms are smaller than one and $\mathbf{P}$ is diagonal matrix with canonical correlations. The $\boldsymbol{\mu}$ corresponds to the sample mean. It is further shown that the posterior expectations of the latent variables lie in the space spanned by the CCA solution.

If $\mathbf{M}_1 = \mathbf{M}_2 = \mathbf{M} = \mathbf{P}^{1/2}\mathbf{R}$, where $\mathbf{R}$ is rotation matrix of size $K$, it can be seen that the probabilistic solution does not in general correspond to the actual projections of CCA. In standard CCA the projections are ordered by the correlation thus identifying the projections up to a sign change. However, Archambeau et al. (2006) provide a method identifying the actual subspace of CCA also for the probabilistic solution.

### 4.3.3 Exponential family canonical correlation analysis

Generalization of CCA to the exponential family is defined as model $p(\mathbf{x}|\boldsymbol{\theta})$, where

$$\begin{aligned}
\boldsymbol{\theta}_1 &= \mathbf{V}_{S1}\mathbf{u}_S + \mathbf{V}_1\mathbf{u}_1 + \boldsymbol{\mu}_1 \\
\boldsymbol{\theta}_2 &= \mathbf{V}_{S2}\mathbf{u}_S + \mathbf{V}_2\mathbf{u}_2 + \boldsymbol{\mu}_2.
\end{aligned}$$

The notation is equivalent to Klami and Kaski (2008). The full model is illustrated in Figure 5, to clarify the role of the various parts of $\mathbf{V}$.

### 4.3.4 Supervised exponential family canonical correlation analysis

Above it was described how ECCA can be used to extract mutual dependencies between two data sets. Besides interpretation tasks, Tripathi et al. (2008) explore how the shared subspace extracted by CCA can be used for classification. They propose to use CCA as a preprocessing method. The method they consider can be seen as supervised CCA where the target information depends only on the shared latent variables. By combining EPLS and ECCA, model for supervised shared components can be built. The novel algorithm is termed supervised ECCA (SECCA).

In the general case for $M$ data sets, where $\mathbf{y}_1$ as in SEPCA (Section 4.1) is used to denote targets, the data vector can be written as

$$\mathbf{x}^T = \begin{pmatrix} \mathbf{y}_1^T & \mathbf{y}_2^T & \dots & \mathbf{y}_M^T \end{pmatrix}.$$

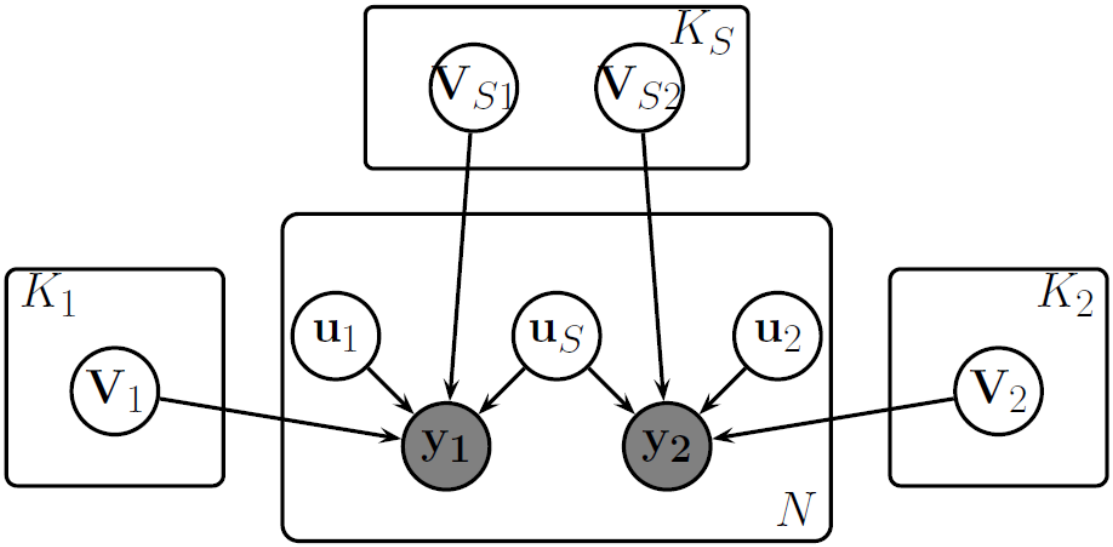



Figure 5: Graphical model for ECCA. Shared variables $\mathbf{u}_S$, $\mathbf{V}_{S1}$ and $\mathbf{V}_{S2}$ capture only mutual dependencies between $\mathbf{y}_1$ and $\mathbf{y}_2$ while set-specific variation for $\mathbf{y}_1$ is modeled with specific variables $\mathbf{u}_1$ and $\mathbf{V}_1$, similarly for $\mathbf{y}_2$. $K$ denotes the ranks of various parts and $N$ refers to the number of observed samples.

The model structure is still written as $\boldsymbol{\theta} = \mathbf{V}\mathbf{u} + \boldsymbol{\mu}$, where

$$\mathbf{u}^T = \begin{pmatrix} \mathbf{u}_S^T & \mathbf{u}_2^T & \dots & \mathbf{u}_M^T \end{pmatrix}.$$

and

$$\mathbf{V} = \begin{pmatrix} \mathbf{V}_{S1} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{V}_{S2} & \mathbf{V}_2 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{V}_{S3} & \mathbf{0} & \mathbf{V}_3 & & \mathbf{0} \\ \vdots & \vdots & & \ddots & \vdots \\ \mathbf{V}_{SM} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{V}_M \end{pmatrix}$$

In SECCA the target-specific variation is left out similarly as in EPLS in order to capture shared predictive features of $\mathbf{y}_1$. To simplify modeling the covariance, the trick of Rish et al. (2008) can be incorporated. That is, different values of $\mathbf{w}$ can be used to determine whether modeling power should be focused on prediction or not. By dropping $\mathbf{y}_1$ the model becomes ECCA for multiple views.

# 5 Priors for exponential family projections

Previously the likelihood functions for EPCA (Section 3) and its various extensions (Section 4) were presented. Following Bayesian inference prior distributions need to be considered defining full joint probability model for the data and the parameters. Setting prior distributions for the parameters is a challenging part of Bayesian modeling and is an active research area. Typically, the choice of prior distribution is a compromise between complex and more realistic priors leading to complicated inference, and simple priors chosen to guarantee efficient computation. Caution must be taken when placing priors to make sure that the resulting posterior is proper, that is, the integral $p(\mathbf{X})$ in (2) does not diverge to infinity.

Most of the research on EPCA-type models has focused solely on ML-solutions or retained prior distributions from the Gaussian Bayesian models. In this thesis, instead, a general prior formulation that takes distribution-specific constraints for the natural parameters into account is proposed.

## 5.1 Background

The first step of Bayesian modeling (see Section 2) is to write down the full probability model for observed and unobserved variables, which in the case of EPCA results in

$$p(\mathbf{u}, \mathbf{V}, \boldsymbol{\mu}, \mathbf{x}|\boldsymbol{\Theta}) = p(\mathbf{x}|\mathbf{u}, \mathbf{V}, \boldsymbol{\mu})p(\mathbf{u}, \mathbf{V}, \boldsymbol{\mu}|\boldsymbol{\Theta}),$$

where $\boldsymbol{\Theta}$ denotes the collection of all the hyperparameters, that is, the parameters of the priors.

The values for $\boldsymbol{\Theta}$ need to set to some suitable values before conditioning on data. However, setting these values is in general hard and one would like to set these values automatically, i.e., learn the values for $\boldsymbol{\Theta}$ from data. One possibility is to treat $\boldsymbol{\Theta}$ as random variables and place so-called hyperprior distribution for $\boldsymbol{\Theta}$. This kind of prior structure is also called hierarchical prior distribution (Gelman et al., 2004). Bayesian inference results in averaging over multiple models that have different $\boldsymbol{\Theta}$ weighted by the hyperprior distribution, although, $\boldsymbol{\Theta}$ can alternatively be determined with cross-validation, as discussed in Section 2.4.

## 5.2 Joint prior

In this thesis a family of prior distributions is presented that incorporates certain common choices as special cases, while being an efficient way of altering the compromise between conjugacy and flexibility in practical models.

Mohamed et al. (2009) extended the EPCA to a full Bayesian model, specifying prior distributions directly for $\mathbf{u}$ and $\mathbf{V}$. This approach is conceptually simple and straightforward, but it is hard to determine which distributions to use. Mohamed et al. (2009) borrowed the assumption of normally distributed latent variables $\mathbf{u}$ from the Gaussian case, while taking $\mathbf{V}$ conjugate to the specific exponential family. The latent variables are assumed to be independent in the prior. Mathematically,

the prior is written as

$$p(\mathbf{U}, \mathbf{V}, \boldsymbol{\mu}|\boldsymbol{\lambda}, \nu, \mathbf{m}, \mathbf{S}, \alpha_0, \beta_0) \propto p(\mathbf{V}|\boldsymbol{\lambda}, \nu)p(\mathbf{U}|\boldsymbol{\mu}, \boldsymbol{\Sigma})p(\boldsymbol{\mu}|\mathbf{m}, \mathbf{S})p(\boldsymbol{\Sigma}|\alpha_0, \beta_0), \quad (32)$$

where, starting from right to left,

$$\boldsymbol{\Sigma}_{ii} \sim i\mathcal{G}(\alpha_0, \beta_0),\ i = 1, \ldots, K\ (\boldsymbol{\Sigma} \text{ is a diagonal matrix}) \quad (33)$$
$$\boldsymbol{\mu} \sim \mathcal{N}(\mathbf{m}, \mathbf{S}) \quad (34)$$
$$\mathbf{u}_n \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}),\ n = 1, \ldots, N \quad (35)$$
$$\mathbf{v}_k \sim Conj(\boldsymbol{\lambda}, \nu),\ k = 1, \ldots, K, \quad (36)$$

where $Conj$ denotes conjugate distribution and $i\mathcal{G}$ denotes the inverse-Gamma distribution. The convenient property of (33) is that fixing the number of latent variables, $K$, is not anymore that critical since unnecessary elements can be driven to zero, that is, $\boldsymbol{\Sigma}_{kk} \approx 0$ with suitable values for $\alpha_0$ and $\beta_0$. Such a prior is generally termed automatic relevance determination (ARD) prior and it is an example of a hierarchical prior (Bishop, 2006).

Unfortunately, (35) is a notoriously bad choice for some exponential family distributions. For example, for the exponential distribution the domain of the natural parameters is the set of strictly positive real numbers, which does not comply with normally distributed $\mathbf{u}$.

In this thesis an alternative novel solution is proposed by imposing the prior on the product of the two variables, instead of formulating separate priors for each. For $\boldsymbol{\theta}_n = \mathbf{V}\mathbf{u}_n + \boldsymbol{\mu}$ it is easy to choose a prior conjugate to the specific exponential family, which takes the correct distribution into account and makes the estimation of $\boldsymbol{\theta}$ easy:

$$\boldsymbol{\theta}_n \sim Conj(\boldsymbol{\lambda}, \boldsymbol{\nu}),\ n = 1, \ldots, N.$$

However, at the same time the connection to the actual factorization is lost; while the model is still parameterized through the low-rank matrices $\mathbf{U}$ and $\mathbf{V}$, the $\mathbf{U}$ and $\mathbf{V}$ are unidentifiable. In practice, this kind of model can still be useful: If the goal is not to analyze the actual components, but merely to find a low-rank approximation of $\mathbf{x}$ (which is sufficient for example for reconstructing the original data from a compressed version), then it is feasible to place the prior directly on $\boldsymbol{\theta}$.

To combine the advantages of the two formulations, (i) separate priors and (ii) the prior for the product, the general prior is introduced

$$p(\mathbf{u}, \mathbf{V}|\boldsymbol{\Theta}, \beta) = \frac{1}{Z} a(\mathbf{V}\mathbf{u})^{\beta} b(\mathbf{u})^{1-\beta} c(\mathbf{V})^{1-\beta}, \quad (37)$$

where $\beta \in [0, 1]$. For clarity, the prior for $\boldsymbol{\mu}$ is dropped from the notation and it is taken to follow $N(\mathbf{0}, \sigma_{\mu}^2\mathbf{I})$ with large $\sigma_{\mu}^2$. The functions $a(\cdot)$, $b(\cdot)$, and $c(\cdot)$ can be arbitrary non-negative functions over the domain of the parameters. The entire normalization is done with $Z$, and hence the functions $a(\cdot)$, $b(\cdot)$ and $c(\cdot)$ need not be normalized. In practice, however, one would typically use standard distributions of the kind used in the above simpler prior assumptions. Then $\beta = 0$ and $\beta = 1$ reduce

the prior into the simpler alternatives, while other values of $\beta$ produce combinations of the two:

$$p(\mathbf{u}, \mathbf{V}|\mathbf{\Theta}, \beta = 0) \propto b(\mathbf{u})c(\mathbf{V}) \tag{38}$$
$$p(\mathbf{u}, \mathbf{V}|\mathbf{\Theta}, \beta = 1) \propto a(\mathbf{V}\mathbf{u}). \tag{39}$$

A useful property of the prior is that if $a(\cdot)$ is set so that it gives zero for values outside the domain of $\boldsymbol{\theta}$, then already a small $\beta$ will be sufficient to restrict the product of individually specified priors for $\mathbf{u}$ and $\mathbf{V}$ to be a legal distribution. This solves the problems of the prior distribution of Mohamed et al. (2009). More generally, the compromise can be thought of as regularization, making the model less sensitive for the specific choices of $b(\mathbf{u})$ and $c(\mathbf{V})$. That considerably simplifies the choice of the distributions, and in practice simple component-wise Gaussian priors,

$$b(\mathbf{u}) = \prod_{k=1}^{K} N(0, \sigma_U^2) \tag{40}$$
$$c(\mathbf{V}) = \prod_{k=1}^{K} \prod_{d=1}^{D} N(0, \sigma_V^2), \tag{41}$$

are used for both, which would not work in general without the regularizing $a(\mathbf{V}\mathbf{u})$ term. The capabilities of the joint prior are demonstrated in Figure 6 for binary data for which the conjugate distribution is the beta distribution. See the caption for more details.

A practical challenge with this kind of a prior is that it is in general known only up to the normalization constant $Z$. This does not pose technical problems with MAP- or MCMC-based inference, since the normalization term cancels out. However, it makes inference on possible hyper-parameters, such as $\sigma_U^2$ and $\sigma_V^2$ above, of the prior difficult. A sampling proposal for the hyper-parameters in the joint prior will need to evaluate the normalization term

$$Z(\mathbf{\Theta}) = \int p(\mathbf{U}, \mathbf{V}|\mathbf{\Theta}, \beta) d\mathbf{\Theta},$$

that, in general, cannot be computed analytically. In this thesis, a simple approach is used; the hyperparameters of $a(\mathbf{V}\mathbf{u})$ are chosen for $\beta = 1$ and for $b(\mathbf{u})$ and $c(\mathbf{V})$ for $\beta = 0$ using a validation set. In the experiments it is shown that already this simple approach leads to a better generalization ability than using either of the extremes.

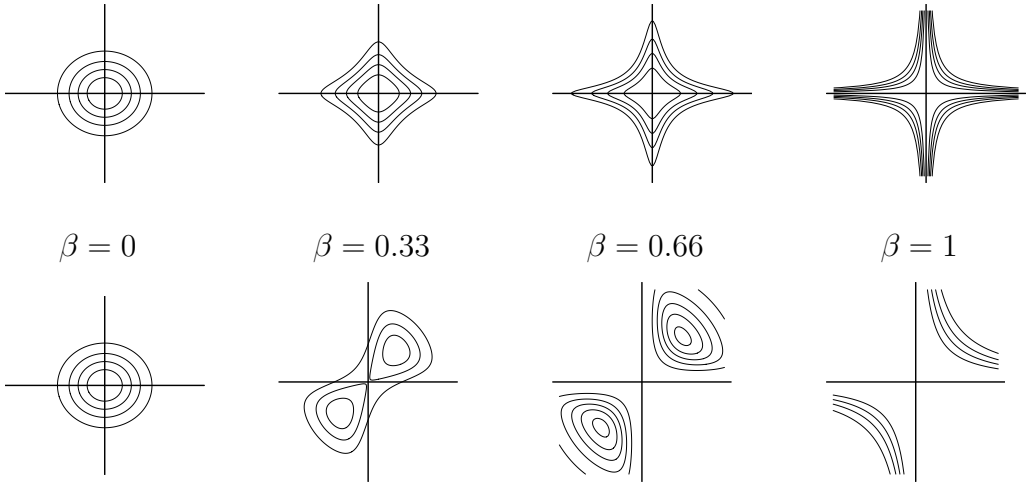



Figure 6: Illustration of the joint prior. The two axes correspond to univariate $u$ and $v$ and the contours plot the probability of $\theta = uv$ under the joint prior, $\ln p(u, v) \propto \beta\big(\lambda uv - g(uv)\big) - \big(1 - \beta\big)\big(1/2u^2 + 1/2v^2\big)$, where $g(x) = \ln(1 + \exp(x))$ with $\ln a(\theta) = \lambda\theta - g(\theta)$. That is, the prior correspond to the beta distribution that is conjugate to the Bernoulli distribution. Additionally we set $\ln b(x) = \ln c(x) = -x^2/(2\sigma^2)$ with $\sigma^2 = 1$. The upper row corresponds to values $\lambda = 10$ and $\nu = 2\lambda$ and the lower to values $\lambda = 0$ and $\nu = 10$. In the lower row negative values for $\theta$ are ruled out with suitably large value for $\beta$. With $\beta = 1$ the prior is improper.

# 6 Inference

After discussing both the likelihood functions and the prior distributions, the next step of Bayesian modeling is to derive the posterior distributions for all of the parameters. Unfortunately, exact inference for the EPCA models is not possible. Two different Markov Chain Monte Carlo (MCMC) sampling methods are discussed that suit different scenarios, but a brief detour on point estimation is first given; while full posterior inference is informative, it may be overkill in some applications.

## 6.1 Point estimates

Point estimates for the model parameters can be inferred from data by maximizing the total log likelihood $\mathcal{L}$ as in (14). Gradient based optimization in the parameter space is adopted to find stationary point of the likelihood, $\nabla\mathcal{L} = 0$, by updating the parameters in the direction of the gradient. MAP estimation is conceptually equally simple; the priors only result in additive terms in $\mathcal{L}$.

The optimization problem in the general case is large, because of the number of variables, and difficult since it is not convex in both arguments. There have been many different proposals for finding point estimates. For Gaussian assumption the latent variables can be marginalized out, significantly decreasing the number of parameters, and the resulting marginal likelihood is maximized (Roweis and Ghahramani, 1999; Tipping and Bishop, 1999).

Guo and Schuurmans (2008) proposed a convex optimization algorithm for the maximum-likelihood case of general exponential family PCA, while MAP estimation requires more generic optimization algorithms. While convex optimization algorithms converge to global optimum, they may be computationally demanding. Collins et al. (2002) present a simple alternating optimization, while Gordon (2002) applies sequential Newton updates. Rish et al. (2008), Schein et al. (2003) and Tipping (2001) use auxiliary functions that are limited to a subset of exponential family distributions, typically Gaussian and Bernoulli distributions. The aim of the auxiliary updates is to make an approximation of the cost function in the neighborhood of the current point such that the gradient of this approximation can be presented in closed form. In this thesis despite all the different optimization methods, conjugate gradients are used following Srebro and Jaakkola (2003). In the experiments the method has turned out to be sufficiently robust algorithm for sensible priors.

## 6.2 Advanced Markov Chain Monte Carlo methods

In Section 2.3 two MCMC algorithms, MH-algorithm and Gibbs sampler were discussed. In this section two efficient samplers suitable for the general exponential family projection models are presented. These are Hybrid Monte Carlo (HMC) and extended Gibbs sampler.

### 6.2.1 Hybrid Monte Carlo sampler

For full Bayesian analysis one can use Hybrid Monte Carlo (HMC) sampler, following Mohamed et al. (2009). Compared to standard MCMC, the Hybrid Monte Carlo (HMC) typically converges faster in large state spaces due to utilizing the gradient information. The EPCA factorization typically has a very large state space, especially in the case of coupled data models where there are separate shared and source-specific latent variables, making HMC a good choice here.

HMC has been applied for large systems such as neural networks (Neal, 1996) and Gaussian processes (Barber and Williams, 1997), and is well described in many text books, see for example (MacKay, 2002). The idea behind HMC is to use molecular dynamic simulation for proposal distribution in the basic MH-algorithm.

HMC is an auxiliary variable sampler. Instead of drawing samples directly from the posterior distribution, the samples are drawn from an augmented distribution $p(\boldsymbol{\psi}, \boldsymbol{t})$ where $\boldsymbol{t}$ is an auxiliary variable and $p(\boldsymbol{\psi})$ is used to denote the distribution from which one wishes to sample. To obtain marginal samples from the augmented distribution, the values for the auxiliary variables $\boldsymbol{t}$ are ignored. In HMC, $\boldsymbol{t}$ corresponds to momemtum variables and the extended target density can be written as

$$p(\boldsymbol{\psi}, \boldsymbol{t}) = p(\boldsymbol{\psi})N(\boldsymbol{t}|\mathbf{0}, \mathbf{I}).$$

HMC first draws the momentum variable $\boldsymbol{t}$ from a Gaussian distribution. Then $L$ 'frog steps' are taken in $\boldsymbol{\psi}$ and $\boldsymbol{t}$ with step size $\tau$. The values of $\boldsymbol{\psi}$ and $\boldsymbol{t}$ after the last step, $L$, are the proposal candidates for the Metropolis-Hastings acceptance step. Pseudo-code for HMC is presented in Algorithm 1, where the frog steps are specified.

The sampler has two parameters, $L$ and $\tau$. In the experiments $L = 10$ and $\tau$ is drawn for each sample from the exponential distribution, following (Neal, 1993). If the step size is too small the simulations are precise but the convergence towards the real posterior distribution can be slow. On the other hand, with large step size more proposals are rejected and hence the chain does not proceed optimally. As $\tau$ is drawn from the exponential distribution $\tau$ is occasionally large, enabling large shifts in the state space, yet most of the time the steps will be small.

### 6.2.2 Identification of components for interpretation

The sampler provides posterior samples for the parameters $\boldsymbol{\Theta}$, and one is typically interested in obtaining marginal distributions of the whole joint posterior, for example for $\mathbf{V}$. As presented in Section 2.3, marginalization can be performed by discarding all other variables that are not of interest. However, there remain severe unidentifiability problems with the proposed models; the marginal expectations of posterior samples are useless. The natural parameters of the models can be represented with $\{\mathbf{U}, \mathbf{V}, \boldsymbol{\mu}\}$ in many different ways,

$$\hat{\mathbf{U}}\hat{\mathbf{V}}^T = (\mathbf{URS})(\mathbf{S}^{-1}\mathbf{R}^T\mathbf{V}^T) = \mathbf{UV}^T,$$

where $\mathbf{R}$ is a unitary rotation matrix and $\mathbf{S}$ is a diagonal scaling matrix. Hence, $\mathbf{U}$ or $\mathbf{V}$ cannot be averaged over the samples, making marginal posterior distributions

**Algorithm 1** Hybrid Monte Carlo

$L$ and $\mu_0$ are parameters of the algorithm. $\mathcal{U}_{[0,1]}$ denotes the uniform distribution on interval $[0,1]$ and $Exp(\lambda)$ the exponential distribution with parameter $\lambda$. The output consists of samples $\boldsymbol{\psi}^{(s)}, s = 1, \ldots, S$.

1. Initialize $\boldsymbol{\psi}^{(0)}$
2. For $s = 0$ to $S - 1$
   3. Sample $a \sim \mathcal{U}_{[0,1]}$, $\boldsymbol{t}^* \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\epsilon \sim Exp(\mu_0)$
   4. Let $\boldsymbol{\psi}_0 = \boldsymbol{\psi}^{(s)}$ and $\boldsymbol{t}_0 = \boldsymbol{t}^* + \epsilon \nabla p(\boldsymbol{\psi}_0)/2$
   5. For $l = 1$ to $L$ take $L$ frog steps
      $\boldsymbol{\psi}_l = \boldsymbol{\psi}_{l-1} + \epsilon \boldsymbol{t}_{l-1}$
      $\boldsymbol{t}_l \ = \boldsymbol{t}_{l-1} + \epsilon_l \nabla p(\boldsymbol{\psi}_l)$
      where $\epsilon_l = \epsilon$ for $l < L$ and $\epsilon_L = \epsilon/2$
   6. If $a < \min\left(1, \frac{p(\boldsymbol{\psi}_L)}{p(\boldsymbol{\psi}^{(s)})} \exp(-\frac{1}{2}(\boldsymbol{t}_L^T \boldsymbol{t}_L - \boldsymbol{t}^{*T} \boldsymbol{t}^*)\right)$
      $\boldsymbol{\psi}^{(s+1)} = \boldsymbol{\psi}_L$
      $\boldsymbol{t}^{(s+1)} = \boldsymbol{t}_L$
      else
      $\boldsymbol{\psi}^{(s+1)} = \boldsymbol{\psi}^{(s)}$
      $\boldsymbol{t}^{(s+1)} = \boldsymbol{t}^*$

difficult to obtain. However, this commonly acknowledged identification problem, also noted by Mohamed et al. (2009) is solved in the following. It should also be kept in mind that even though the components of the product do not identify, the product $\mathbf{U}\mathbf{V}^T$ itself does, and $\mathbf{U}\mathbf{V}^T$ is already sufficient for predictions. Hence, this unidentifiability issue is a problem only for interpretation of the components.

Denote $\boldsymbol{\Theta} = \mathbf{U}\mathbf{V}^T + \mathbf{1}\boldsymbol{\mu}^T$ as the matrix of natural parameters. Decompose $\boldsymbol{\Theta}$ with PCA (Sect. 3.2). Mathematically the decomposition can also be written as

$$\boldsymbol{\Theta} = \widehat{\mathbf{U}} \boldsymbol{\Sigma} \widehat{\mathbf{V}}^T,$$

where $\boldsymbol{\Sigma} = diag(\mu_1^2 \ldots \mu_D^2)$ and $\widehat{\mathbf{V}} = \begin{pmatrix} \hat{\mathbf{v}}_1 & \ldots & \mathbf{v}_D \end{pmatrix}$ define the eigenvalues and eigenvectors of $\boldsymbol{\Theta}^T \boldsymbol{\Theta} \hat{\mathbf{v}}_i = \mu_i^2 \hat{\mathbf{v}}_i$. Defining $\widehat{\mathbf{U}} = \boldsymbol{\Theta} \widehat{\mathbf{V}} \boldsymbol{\Sigma}^{-1}$ it can be further verified that $\boldsymbol{\Theta} = \widehat{\mathbf{U}} \boldsymbol{\Sigma} \widehat{\mathbf{V}}^T = \boldsymbol{\Theta} \widehat{\mathbf{V}} \boldsymbol{\Sigma}^{-1} \boldsymbol{\Sigma} \widehat{\mathbf{V}}^T = \boldsymbol{\Theta}$.

One can average over the decomposed matrices made for $S$ posterior samples $\boldsymbol{\Theta}^{(s)} = \widehat{\mathbf{U}}^{(s)} \boldsymbol{\Sigma}^{(s)} \widehat{\mathbf{V}}^{(s)T}$, where $s = 1, \ldots, S$ and use a simple method to correct for the sign unidentifiability, i.e. $\mathbf{U}\mathbf{V}^T = (-\mathbf{U})(-\mathbf{V}^T)$, of PCA. Collect the $i$th column of $\widehat{\mathbf{U}}^{(s)}$, denoted as $\widehat{\mathbf{U}}_i^{(s)} \in \mathbb{R}^N$, in a matrix $\mathbf{A} = \begin{pmatrix} \mathbf{U}_i^{(1)} & \mathbf{U}_i^{(2)} & \ldots & \mathbf{U}_i^{(S)} \end{pmatrix}^T$. For the matrix $\mathbf{A}$ run K-means clustering algorithm (Bishop, 2006) setting $K = 2$. This is a very simple clustering task as the two cluster centers are $\mathbf{c}$ and $-\mathbf{c}$. Finally, change the sign of samples in either cluster. The process is repeated for $i = 1, \ldots, D$. Similar process is performed for the columns of $\widehat{\mathbf{V}}$. Unfortunately, in the case of any $\mu_i = \mu_j$ for $i \neq j$ the vectors $\hat{\mathbf{v}}_i$ and $\hat{\mathbf{v}}_j$ cannot be determined. See Figure 7 for an intuitive explanation.

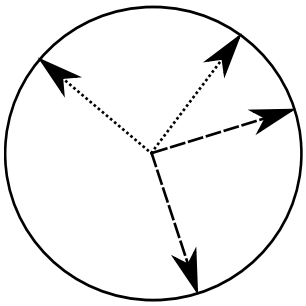


Figure 7: The directions of two eigenvectors cannot be determined if the corresponding eigenvalues are equivalent. The differently dashed pairs of vectors correspond to the possible eigenvectors. The sphere illustrates a 2-dimensional subspace where the eigenvalues are equivalent. The direction of the first vector is random while the second has to be orthogonal to the first one.

An alternative, computationally less demanding method, is to average first over different $\boldsymbol{\Theta}^{(s)}$ obtaining $\widehat{\boldsymbol{\Theta}} = \frac{1}{S}\sum_{s=1}^{S}\boldsymbol{\Theta}^{(s)}$ and decompose the resulting single matrix. This method is suitable for visualization applications since it suffices to plot the posterior means.

For Bayesian EPLS (BEPLS) and Bayesian ECCA (BECCA) one is usually only interested in the shared components. Hence, only $\boldsymbol{\Theta}_S = \mathbf{U}_S\mathbf{V}_S$ corresponding to the shared components is decomposed.

In principle, it is also possible to add constraints, such as $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, as in classical PCA, in the sampler. Hoff (2007) showed that sampling with such constraints for $\mathbf{U}$ and $\mathbf{V}$ is possible, but points out that sampling can be painfully slow.

### 6.2.3 Extended Gibbs sampler

For the Gaussian Bayesian CCA (Klami and Kaski, 2007) analytic marginalization of the source-specific variables is possible and the rotational ambiguity can be solved revealing the true canonical projections and the corresponding correlations (Section 4.3). In ECCA, however, it is hard to make sure the modeling power is divided correctly between the $\mathbf{u}_S$, $\mathbf{u}_1$, and $\mathbf{u}_2$. In principle the sampler explores the space of solutions correctly, but the convergence may be slow and the sampler may not mix well.

In this thesis, a novel sampler that utilizes the more efficient solutions for the Gaussian models as part of the sampler for the general exponential family is introduced. The practical sampling algorithm, coined GiBECCA, proceeds by alternating two separate sampling steps. The algorithmic approach, explained next, is similar to how Hoff (2007) made inference for binary PCA. The intuitive idea is to alternate between two sampling stages. In one stage, the $\boldsymbol{\Theta}$ is treated as data that *a priori* follows normal distribution, and learn a factorization $\boldsymbol{\Theta} = \mathbf{U}\mathbf{V}^T$ for that. The other stage then updates $\boldsymbol{\Theta}$, taking into account the exponential family likelihood and the actual data $\mathbf{X}$.

After initialization (the first step of Algorithm 2), new values for the parameters

**Algorithm 2** GiBECCA

Hyperparameters for the initialization need to be set. $i\mathcal{G}$ denotes inverse-gamma distribution and $i\mathcal{W}$ denotes inverse-Wishart distribution. Notation $\tilde{\boldsymbol{\theta}}$ denotes centered variables.

1. Initialization, $i = 1, \ldots, K$ and $n = 1, \ldots, N$
   $\mathbf{v}_i \sim \mathcal{N}(\mathbf{0}, \gamma_i \mathbf{I})$
   $\gamma_i \sim i\mathcal{G}(\alpha_0, \gamma_0)$
   $\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2 \sim i\mathcal{W}(\mathbf{S}_0, \nu_0)$
   $\boldsymbol{\mu} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$
   $\mathbf{u}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
   $\boldsymbol{\theta}_n \sim \mathcal{N}(\boldsymbol{\mu} + \mathbf{V}\mathbf{u}_n, \boldsymbol{\Psi})$, where $\boldsymbol{\Psi} = \begin{pmatrix} \boldsymbol{\Psi}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Psi}_2 \end{pmatrix}$
2. For $s = 0$ to $S - 1$
   3. Sample parameters
      $\gamma_i | \mathbf{v}_i \sim i\mathcal{G}(\frac{1}{2}D + \alpha_0, \frac{1}{2}\mathbf{v}_i^T \mathbf{v}_i + \gamma_0)$
      $\boldsymbol{\Psi}_j |\ \tilde{\boldsymbol{\Theta}}_j, \mathbf{V}_{Sj}, \mathbf{U} \sim i\mathcal{W}(\mathbf{S}_0 + \mathbf{S}, \nu_0 + N)$,
      where $\mathbf{S} = \sum_n (\tilde{\boldsymbol{\theta}}_{jn} - \mathbf{V}_{Sj}\mathbf{u}_n)(\tilde{\boldsymbol{\theta}}_{jn} - \mathbf{V}_{Sj}\mathbf{u}_n)^T$
      $\boldsymbol{\mu} | \boldsymbol{\Theta}, \boldsymbol{\Psi}, \mathbf{U} \sim \mathcal{N}(\boldsymbol{\Sigma}(\boldsymbol{\Psi} + \mathbf{V}\mathbf{V}^T)^{-1} \sum_n \boldsymbol{\theta}_n, \boldsymbol{\Sigma})$,
      where $\boldsymbol{\Sigma}^{-1} = N(\boldsymbol{\Psi} + \mathbf{V}\mathbf{V}^T)^{-1} + \frac{1}{\sigma^2}\mathbf{I}$
      $\mathbf{v}_k | \tilde{\boldsymbol{\Theta}}, \mathbf{V}_{-k}, \mathbf{U}, \boldsymbol{\Psi}, \gamma_k \sim \mathcal{N}(\boldsymbol{\Psi}^{-1}\boldsymbol{\Sigma} \sum_j \mathbf{U}_{kj}(\tilde{\boldsymbol{\theta}}_j - \mathbf{V}_{-k}\mathbf{U}_{-kj}), \boldsymbol{\Sigma})$
      where $\boldsymbol{\Sigma} = \sum_j \mathbf{U}_{kj}^2 \boldsymbol{\Psi}^{-1} + \frac{1}{\gamma_k}\mathbf{I}$
      $\mathbf{u}_n | \tilde{\boldsymbol{\theta}}_n, \mathbf{V}, \boldsymbol{\Psi} \sim \mathcal{N}(\mathbf{V}^T \boldsymbol{\Sigma} \tilde{\boldsymbol{\theta}}_n, \mathbf{I} - \mathbf{V}^T \boldsymbol{\Sigma} \mathbf{V})$,
      where $\boldsymbol{\Sigma}^{-1} = \mathbf{V}\mathbf{V}^T + \boldsymbol{\Psi}$
   4. Sample
      $\boldsymbol{\theta}_n^* | \sim \mathcal{N}(\mathbf{V}\mathbf{u}_n + \boldsymbol{\mu}, \boldsymbol{\Psi})$
      accept $\boldsymbol{\Theta}_{ij}^*$ with probability $\min\left(1, \frac{p(\mathbf{X}_{ij} | \boldsymbol{\Theta}_{ij}^*)}{p(\mathbf{X}_{ij} | \boldsymbol{\Theta}_{ij})}\right)$

are drawn using the Gibbs sampler for the Gaussian BCCA (Klami and Kaski, 2007) treating $\boldsymbol{\Theta}$ as data (step 3 of Algorithm 2). Next, given the actual data $\mathbf{X}$ and the current values for the parameters, the natural parameter matrix $\boldsymbol{\Theta}^*$ is sampled using the Metropolis-Hastings applying the predictive distribution of Gaussian BCCA as the proposal distribution (step 4 of Algorithm 2). The algorithm proceeds by alternating steps 3 and 4 until convergence.

The sampler is efficient but approximative; as evident from the procedure, it requires $\beta = 0$ (in Equation (37)) and Gaussian priors for both $\mathbf{u}$ and $\mathbf{V}$. Explicit normality assumption for $\boldsymbol{\theta}$ may violate distribution-specific assumptions for the natural parameters as can be seen from the Table 1. Such restrictions apply, for example, for the exponential distribution; additional parameter transformation for $\boldsymbol{\theta}$ or adjustment of the priors may be needed. In the experiments it is studied how restrictive the normality assumption for the natural parameters is in practice.

# 7 Experiments and results

The experimental section consists of four separate experiments. First, supervised exponential family principal component analysis (SEPCA) is compared to exponential family partial least squares (EPLS) in prediction tasks, where the covariate information is known to contain source-specific noise components. Second, the effect of the proposed joint prior for the EPCA model is studied in missing value imputation task.

The two remaining experiments proceed with Bayesian exponential family canonical correlation analysis (BECCA). In the first experiment it is demonstrated why the correct data type needs to be taken into account by showing that the correct assumption leads to better accuracy in a classification task. Additionally the proposed sampling algorithms are compared. Finally, a demonstration of BECCA for analyzing the shared dependencies between movie genres and descriptions is given by visualizing the posterior distributions of the first two canonical projections.

## 7.1 Supervised dimensionality reduction

The first empirical experiment shows the importance of separately modeling the data-specific noise in supervised learning. Using artificial toy data, the difference between supervised EPCA (SEPCA) and the exponential family PLS (EPLS) is demonstrated.

Binary data is created from the model (27) with $K_S = 1$, $K_2 = 5$ and $D_1 = 1$ and $D_2 = 20$. 50 samples are used for training and 950 for testing. The inference is performed with conjugate gradient method (Section 6.1) and the models are compared in the task of predicting $\mathbf{y}_1$ for the left-out testing samples, using prediction error, the proportion of misclassified samples, as the performance measure. The results are averaged over 80 realizations of randomly generated data sets.

As the shared source is only one-dimensional, it is possible to reach maximal prediction accuracy already with one component. However, SEPCA with just one component does not find the true solution as it is confused by the noise specific to $\mathbf{y}_2$. The model will still reach the optimal prediction accuracy, but requires 6 components to do it (Figure 8). The trick of Rish et al. (2008) in (26), lowering the importance of modeling $\mathbf{y}_2$, helps by improving the predictive performance for a low number of components, but for optimal performance still as many components are needed.

EPLS, however, finds the true solution of one-dimensional shared space, while modeling all the source-specific noise with separate components. Hence, it achieves the same predictive performance already with a single component. The difference in performance is statistically significant for $1 - 5$ components ($p < 0.05$ t-test with Bonferroni correction). It is worth noting that the computational load of the complexities sufficient for optimal prediction is comparable; SEPCA requires 6 components, while EPLS requires 1 shared and 5 noise components. The added benefit of EPLS is primarily improved interpretation: one can directly tell that the interesting subspace is one-dimensional.

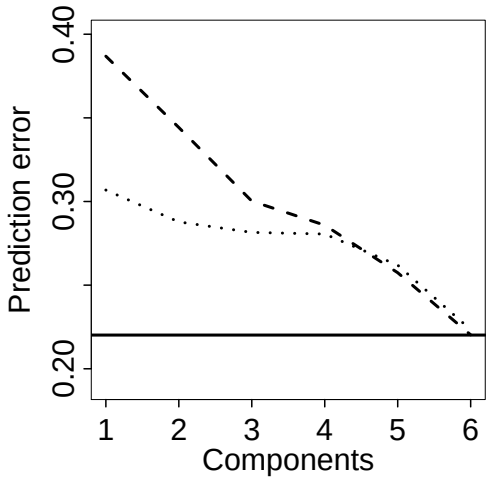



Figure 8: Prediction errors (lower is better) for the supervised EPCA experiment. The solid line depicts the error for a one-component EPLS-solution, while the other two curves are classical SEPCA models. The dashed line assumed equal modeling power for the target and covariates, while the dotted line weights the covariate modeling part with $\alpha = 10^{-3}$. A wide range of values of $\alpha$ result in similar performance (not shown).

## 7.2 The effect of the prior

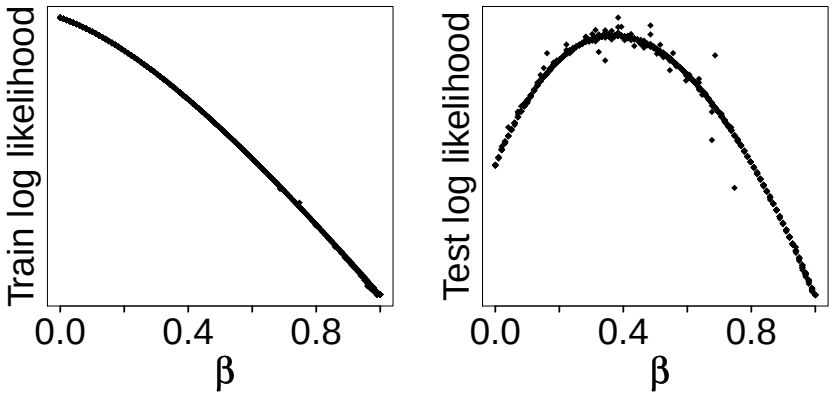
In Section 5, a family of prior distributions controlled by the regularization parameter $\beta$ was presented. Here it is illustrated how the regularization improves the predictive performance of the model on the SPECT data obtained from the UCI repository (`http://archive.ics.uci.edu/ml/datasets/SPECT+Heart`). The standard PCA task of missing value imputation with one component is solved for 100 values of the regularization parameter, and the reconstruction performance is measured by the log likelihood (14).

As the data is binary, the Bernoulli distribution and prior

$$a(\mathbf{U}\mathbf{V}^T) = \prod_{n=1}^{N}\prod_{d=1}^{D} \lambda(\mathbf{U}\mathbf{V}^T)_{nd} - \nu g\Big((\mathbf{U}\mathbf{V}^T)_{nd}\Big),$$

with Gaussian priors for $\mathbf{U}$ and $\mathbf{V}$ (as in Equations (40) and (41)) are chosen. That is, the computationally simple assumption of individual Gaussian priors for $\mathbf{U}$ and $\mathbf{V}$ is made. This does not necessarily match the requirements of the Bernoulli distribution and that is why the prior is complemented with a separate conjugate term for $\mathbf{U}\mathbf{V}^T$, which is specifically chosen for the observation likelihood in order to regularize the solution.

The main purpose of the experiment is to illustrate the effect of the regularization parameter $\beta$. The hyperparameters are chosen with a validation set separately for $\beta = 0$ and $\beta = 1$ resulting in values $\sigma_V^2 = 100$ and $\sigma_U^2 = 0.001$ and $\lambda = 0.1$ and $\nu = 0.2$. The $\beta$ parameter is then varied, keeping the hyperparameters fixed, and




(a) (b)

Figure 9: Illustration of the reconstruction performance with different regularization values. For each $\beta$ 10 random initializations are used and all of the results are included in the plot to illustrate how already the simple conjugate gradient algorithm almost always converges to the the global optimum. The best generalization ability is obtained with intermediate value demonstrating that the full prior outperforms the simpler ones.

in Figure 9 it is shown that the optimal predictive performance is obtained with $\beta$ around 0.4. That is, regularizing a sensible model with separate priors $p(\mathbf{U})p(\mathbf{V})$ by conjugately defined prior on $\mathbf{U}\mathbf{V}^T$ improves the predictive performance.

## 7.3 Exponential family canonical correlation analysis

### 7.3.1 Classification in the joint space

One use for CCA is in finding a shared representation that contains the variation relevant to both data sources. The ability to do that can be indirectly measured by attempting to classify the samples given the representation (Tripathi et al., 2008). On artificial data where the shared variation is known to be predictive of the class labels, a model extracting the true shared variation should have the best performance.

Two collections of toy data sets are created from the model in Section 4.3.4 for $M = 3$ with $K_S = 1$, $K_2 = 2$ and $K_3 = 2$ and $N = 50$, $D_1 = 1$ and $D_2 = D_3 = 20$. CCA is only ran for $\mathbf{y}_2$ and $\mathbf{y}_3$ using $\mathbf{y}_1$ as external label set used to measure the accuracy of the different methods. The first collection is binary and the second is count data. Four different variants of CCA are learned to study the effect of the link function and inference algorithm. First, standard linear CCA is applied to obtain a baseline. As shown in Figure 10, it overfits severely to such small data for both distributions. Bayesian Gaussian CCA (Klami and Kaski, 2007), which has an incorrect link function here (namely the identity function), outperforms classical CCA on binary data, but not on the skewed data with Poisson distribution.

The exponential family CCA with correct distributional assumptions and HMC inference outperforms the alternatives for both data sets, showing the importance of

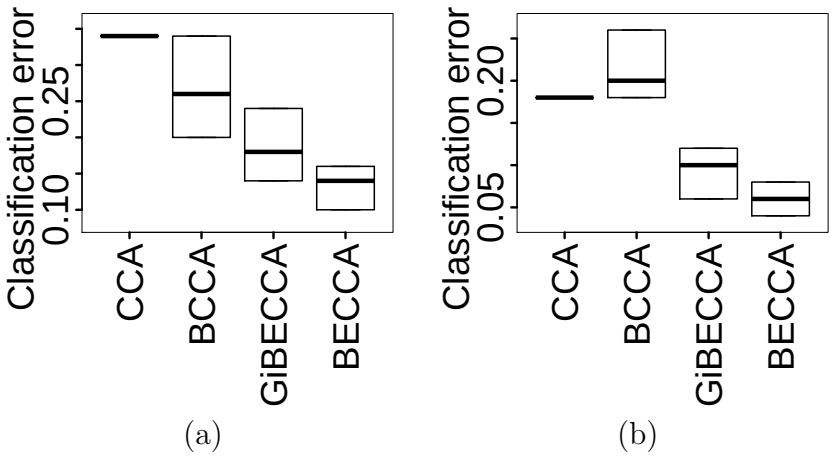



Figure 10: Performance of various CCA models, measured as the classification error of a K-nearest neighbor classifier with $K = 9$ in the shared latent space. All Bayesian variants outperform classical CCA for binary data (a), and the exponential family variants making the correct distribution assumption outperform the one with incorrect distribution (Gaussian; BCCA), especially for the more skewed Poisson data (b). The two sampling algorithms for the exponential family variant are comparable, but the HMC sampler with $\beta = 0.5$ (BECCA) gives slightly better accuracy than GiBECCA. The box-plots show the 25%, 50% and 75% quantiles.

using the right distribution. Here the regularization value was fixed to $\beta = 0.5$ and we used $\lambda = 0.1$ and $\nu = 0.2$ for the hyperparameters. Finally, for comparison the alternative sampler presented in Section 6.2.3 (GiBECCA) is included. It is slightly worse than the HMC sampler using the full prior of (37), but has the advantage of faster convergence (Figure 11).

### 7.3.2 Movie data

To demonstrate the visualization capabilities of ECCA, a small collection of movies described with two views, selected from information available in the Allmovie database (`http://www.allmovie.com/`) are analyzed. The first view is the binarized bag-of-words representation of a brief description of the movie, while the other is a multivariate genre classification in binary format. Each movie may belong to a subset of 10 genres, which extends the task beyond supervised visualization or SPCA.

The main interest is in demonstrating the capability of BECCA to separate shared information from structured "noise" present in only one of the views. Hence, the content descriptions are manually constructed to contain both. A subset of terms (total of 32 terms listed in Table 3) for the bag-of-words representation is manually chosen, so that half of the terms were chosen as genre-related and half were other terms chosen near the genre-related terms in frequency order to provide a contrast group. As an example, the most frequent terms in the genre-related set are *love*, *comedy* and *drama*, while the corresponding words in the noise set are *two*, *woman*, *some*, chosen because their frequency matched best the genre-related words.

BECCA is applied on this data, aiming to extract the components that best

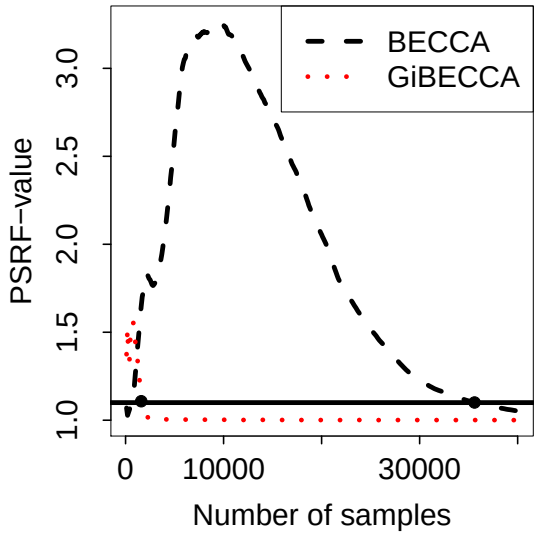
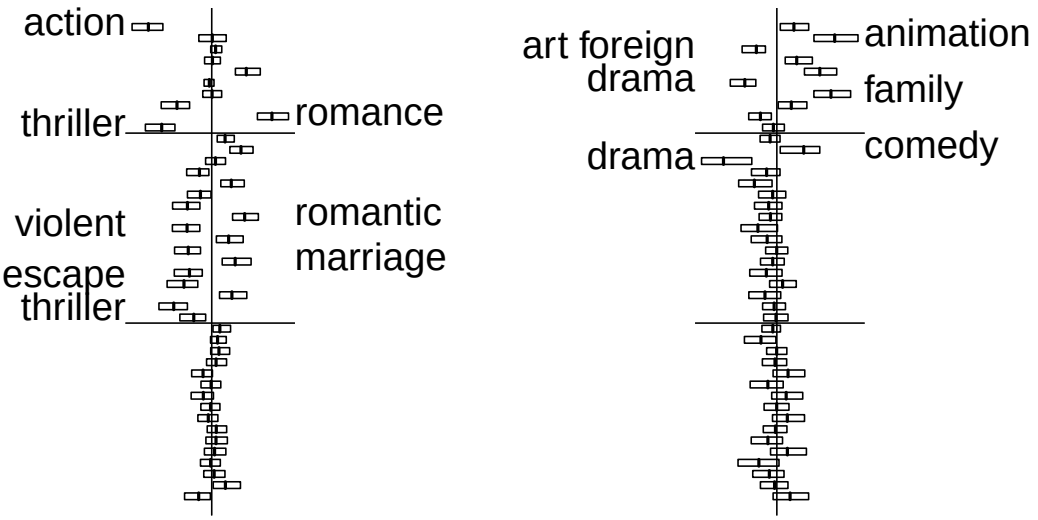



Figure 11: Comparison of the sampling methods. The convergence of the two samplers is monitored by calculating the PSRF-values (Section 2.3) with 4 separate chains. The horizontal line corresponds to value 1.1 and the chain is converged if the value falls below that (Gelman et al., 2004). GiBECCA converges considerably faster than BECCA (approximately 20-fold difference). The figure also demonstrates that if PSRF value is monitored too early it can give misleading results, because the movement of the BECCA sampler in the beginning is slow (PSRF values smaller than 1.1).

capture the genre variation. The GiBECCA inference method (Algorithm 2) is used because interpreting the actual components is easier in that model, and because it was found out to be nearly as good as HMC with the full prior in the previous experiment while being much faster to compute. Figure 12 shows the first two shared projection vectors, that is, the first two columns of $\mathbf{V}$. It is immediately noticed how the part covering the noise-terms in $\mathbf{V}_{S2}$ is around zero for all terms, showing that the shared components do not capture description-specific noise. At the same time, each projection picks a subset of genre-related terms and actual genre memberships. Closer inspection of the features reveals that the first component separates romantic movies from action movies, while the second component mainly separates family-targeted genres (cartoons, family movies) from drama.

Table 3: Features of movie data set.

| Genres | Genre-related descriptions | Genre-independent descriptions |
|---|---|---|
| action | love | two |
| animation | comedy | woman |
| art foreign | drama | some |
| classic | death | brother |
| comedy | relationship | now |
| drama | war | former |
| family | police | little |
| horror | romantic | still |
| romance | violent | head |
| thriller | beautiful | involved |
| | action | town |
| | marriage | keep |
| | crime | few |
| | escape | working |
| | romance | offer |
| | thriller | meanwhile |




(a) (b)

Figure 12: Illustration of the first two CCA components of the movie data. In both figures the top 10 bars represent the 10 genre membership indicators, the next 17 bars the genre-related words in the textual description of the movie, and the bottom 16 bars the genre-independent terms. Genre-related terms are present in the projections much more strongly than the genre-independent 'noise'-terms, as they should. The first shared component (a) picks most genre-related terms, detecting a strong link between the genre memberships and the descriptions. The second shared component (b) extracts a more detailed relationship: family/comedy/animation movies are separated from the rest by absence of the word *drama* in the descriptions.

# 8 Discussion

Exponential family generalization of principal component analysis (EPCA) is an active research area with recent publications published in the top machine learning conferences (Collins et al., 2002; Gordon, 2002; Guo and Schuurmans, 2008; Guo, 2009; Mohamed et al., 2009; Sajama and Orlitsky, 2004; Schein et al., 2003; Singh and Gordon, 2008; Rish et al., 2008; Tipping, 2001). In this thesis, a general framework for matrix factorizations or projection methods in the exponential family was presented by building on EPCA. It was described how various methods for analyzing paired data sources can be derived from the general model by simple restrictions on the projection vectors. As practical examples the first Bayesian exponential family variants of partial least squares and canonical correlation analysis were presented, opening up possibilities for applying basic tools for varying data types in a principled way. It was also shown how straightforward prior assumptions may lead to poor performance in exponential family models, and a regularizing prior was given to overcome the problems. Furthermore, inference methods were discussed for the models demonstrating the variants on both artificial and real life data. The new sampling algorithm presented in this thesis, was shown to be much faster than the previous approach (Mohamed et al., 2009) but slightly less accurate.

However, there is still work to be done: There are not yet good methods for inferring hyperparameters of the regularizing prior (which makes also model complexity selection difficult), and there remain open challenges in how to effectively do inference for CCA-like models with explicit noise components. For Gaussian models the inference is efficient due to analytical marginalization, whereas other distributions are still lacking efficient algorithms. A solution that utilizes the marginalized variant as a part of the process was proposed, but it is only applicable for a special case of our general prior family.

# References

Archambeau, C. and Bach, F. (2009). Sparse probabilistic projections. In *Advances in Neural Information Processing Systems 21*, pp. 73–80, Cambridge, MA, MIT Press.

Archambeau, C., Delannay, N. and Verleysen, M. (2006). Robust probabilistic projections. *Proceedings of the 23rd International Conference on Machine Learning*, pp. 33–40, New York, ACM press.

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

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

Barber, D. and Williams, C. K. (1997). Gaussian processes for Bayesian classification via hybrid Monte Carlo. In: *Advances in Neural Information Processing Systems 9*, pp. 340-346, Cambridge, MA, MIT Press.

Bingham, E., Kaba¡n, A. and Fortelius, M. (2009). The aspect Bernoulli model: multiple causes of presences and absences. Pattern Analysis Applications, 12(1):55-78

Bishop, C.M. (2006). Pattern recognition and machine learning. Springer, New York.

Bernardo, J.M. and Smith, A.F (2000). Bayesian theory, Wiley, Chichester.

Bo, L. and Schimsesku, C (2009). Supervised spectral latent variables models. In *Proceedings of the 12th Conference on Artificial Intelligence and Statistics*, Florida, USA, JMLR:W&CP 5:33-39.

Collins, M., Dasgupta, S., and Schapire, R. E. (2002). A generalization of principal components analysis to the exponential family. In *Advances in Neural Information Processing Systems 14*, pp. 617–624. Cambridge, MA, MIT Press.

Caruana, R. (1997). Multitask learning. Machine Learning, 28:41-75.

Gelman, A., Carlin, J.B., Stern, H.S. and Rubin, D.B. (2004). Bayesian data analysis (2*nd edition*). Chapman & Hall, Boca Raton FL.

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

Gordon, G. (2002). Generalized$^2$ linear$^2$ models In *Advances in Neural Information Processing Systems 15*, pp. 577–584. Cambridge, MA, MIT Press.

Guo, Y. (2009). Supervised exponential family principal component analysis via convex optimization. In *Advances in Neural Information Processing Systems 21*, pp. 569–576. Cambridge, MA, MIT Press.

Guo, Y. and Schuurmans, D. (2008). Efficient global optimization for exponential family PCA and low-rank matrix factorization. In *Allerton Conference on Communications, Control, and Computing*, pp. 1100–1107. IEEE.

Gustafsson, M. G. (2001). A probabilistic derivation of the partial least-squares algorithm. *Journal of Chemical Information and Modeling*, 41:288–294.

Heller, K.A., Williamson, S. and Ghahramani, Z. (2008). Statistical models for Partial membership. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 392–399, New York, ACM.

Hoff, P. D. (2007). Model averaging and dimension selection for the singular value decomposition. *Journal of the American Statistical Association*, 102:674–685.

Hoff, P. D. (2007). Simulation of the matrix Bingham-Von Mises-Fisher distribution with applications to multivariate and relational data, http://arxiv.org/pdf/0712.4166v1.

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

Hotelling, H. (1936). Relations between sets of variates. Biometrika, 38:321–377.

Jolliffe, I.T. (1986). Principal component analysis. Springer-Verlag, New York.

Klami, A. and Kaski, S. (2007). Local dependent components. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 425–432. Omnipress, Madison, WI.

Klami, A. and Kaski, S. (2008). Probabilistic approach to detecting dependencies between data sets. *Neurocomputing*, 72:39–46.

Lawrence, N. (2005). Probabilistic non-linear principal component analysis with Gaussian process latent variable models. Journal of Machine Learning Research, 6:1783–1816.

Lee, D.D. and Seung, H.S. (2001). Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13*, pp.556–562, Cambridge, MA, MIT press.

Lian, H. (2009). Bayesian nonlinear principal component analysis using random fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31: 749–754.

Ma H., Yang H., Lyu M.R. and King I. (2008). SoRec: Social recommendation using probabilistic matrix factorization Proceedings of 17th Conference on Information and Knowledge Management, pp. 931–940, New York, ACM.

MacKay, D.J. (2002). Information theory, inference & learning algorithms. Cambridge University Press.

Mohamed, S., Heller, K., and Ghahramani, Z. (2009). Bayesian exponential family PCA. In *Advances in Neural Information Processing Systems 21*, pp. 1089–1096. Cambridge, MA, MIT Press.

Neal R.M. (1993). Probabilistic inference using Markov Chain Monte Carlo methods. Tech. Report CRG-TR-93-1,University of Toronto, Department of Computer Science.

Neal, R. M. (1996). Bayesian learning for neural networks. Springer-Verlag, New York.

Nounou, M., Bakshi, B., Goel, P., and Shen, X. (2002). Process modeling by Bayesian latent variable regression. *AIChE Journal*, 48:1775–1793.

Pearson, K. (1901). On lines and planes of closest fit to the systems of points in space. Philosophical Magazine, 2:559-572.

Rish, I., Grabarnik, G., Cecchi, G., Pereira, F., and Gordon, G. J. (2008). Closed-form supervised dimensionality reduction with generalized linear models. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 832–839, New York, NY, ACM.

Roweis, S. and Ghahramani, Z. (1999). A unifying review of linear Gaussian models. *Neural Computation*, 11:305–345.

Sajama and Orlitsky, A. (2004). Semi-parametric exponential family PCA. In *Advances in Neural Information Processing Systems 17*, pp. 1177–1184, Cambridge, MA, MIT Press.

Salakhutdinov R. and Mnih A. (2008). Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems 20*. Cambridge, MA, MIT Press.

Salakhutdinov R. and Mnih A. (2008) (b). Bayesian probabilistic matrix factorization using Markov Chain Monte Carlo. In*Proceedings of the 25th international conference on Machine learning*, pp. 880-887, New York, ACM.

Schein A.I, Saul L.K., and UngarL.H. (2003). A generalized linear model for principal component analysis of binary data. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*,

Shawe-Taylor, J. and Cristianini, N. (2004). Kernel methods for pattern recognition, Cambridge University Press.

Singh, A.P. and Gordon, G.J. (2008). Relational learning via collective matrix factorization. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.650-658, New York, ACM.

Srebro, N. and Jaakkola, T. (2003). Weighted low-rank approximations. In *Proceedings of the 20th International Conference on Machine Learning*, pp. 720-727. USA, AAAI Press.

Tipping, M.E. (2001). Probabilistic visualization of high-dimensional binary data In *Neural Information Processing Systems 11*, pp.592-598, Cambridge, MA, MIT Press.

Tipping, M. and Bishop, C. (1999). Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11:443–482.

Tripathi, A.., Klami, A., and Kaski, S. (2008). Simple integrative preprocessing preserves what is shared in data sources. BMC Bioinformatics 2008, 9:111

Vinokourov, A., Christianini, N., and Shawe-Taylor, J. (2003). Inferring a semantic representation of text via cross-language correlation analysis. In *Advances in Neural Information Processing Systems 15*, pp. 1473–1480. Cambridge, MA, MIT Press.

Williamson, S and Ghahramani, Z. (2008). Probabilistic models for data combination in recommender systems. In *Advances in Neural Information Processing Systems 20 Workshop: Learning from Multiple Sources*

Yu, S., Yu, K., Tresp, V., Kriegel, H.-P., and Wu, M. (2006). Supervised probabilistic principal component analysis. In *Proceedings of the 30th annual International ACM SIGIR Conference on Knowledge Discovery and Data Mining*, pp. 464–473. New York, NY, ACM Press.