Context assisted information extraction

A Thesis

Presented to the School of Computing

University of the West of Scotland

In Partial Fulfilment

of the Requirements for the Degree of Doctor of Philosophy

By

Gayle Leen

Applied Computational Intelligence Research Unit

University of the West of Scotland, High Street, Paisley PA1 2BE, Scotland

Email: gayle.leen@uws.ac.uk

Abstract

While there are numerous unsupervised learning methods in the machine learning literature for exploring the structure of a *single* data set, less attention has been paid to the unsupervised learning of multiple data sets that have a shared structure. In this thesis, we show how to handle this problem in a probabilistic generative framework, limiting our analysis to the case of two related data sets. Each data set acts as a *context* to guide the feature extraction for the other.

Chapter 2 presents the background to probabilistic modelling, dimensionality reduction techniques, and existing methods for exploring two related data sets. Based on an information theoretic analysis of the dependencies between two related data variables, in Chapters 3 and 4 we develop two generative models based on the Gaussian Process Latent Variable Model (GPLVM), providing a probabilistic interpretation of nonlinear canonical correlation analysis. In Chapter 5, a mixture of probabilistic canonical correlation analysers is used to model two data sets that are nonlinearly related to a shared latent space. We then show how to overcome the problem of determining the number of mixture components, through a fully Bayesian treatment of the model. A Dirichlet process prior is placed on the indicator variables, allowing an infinite number of components, such that the number of *represented* components is inferred automatically.

Acknowledgements

I would like to thank my supervisor Colin Fyfe for his enthusiasm and guidance while I was studying for my PhD. Also, I would like to thank the other researchers I met while at the University of the West of Scotland, including Marian Peña, Andreas Loengarov, Iain Miller, Wesam Barbakh, and Benoit Chaperot. Additional thanks to Ata Kabán, who I visited at the University of Birmingham, and Neil Lawrence and Carl Ek, who I visited at the University of Manchester. Thanks to Rik Fransens for useful discussions about all things Bayesian. Finally, I would like to thank Jos Koetsier, and my family including my dad, who sadly passed away in October 2005.

This work was carried out in the Applied Computational Intelligence Unit at the University of the West of Scotland, and funded by a studentship from the School of Computing.

Contents

Chapter 1

Introduction

Humans have to continually make decisions based on sensory observations of an uncertain and changing environment, and learn to adapt their behaviour according to the observations. The learning process can be thought of as creating a *model* of a set of observed data, with the aim of making predictions about future observations. This characterisation of learning is the basis of the field of machine learning. Machine learning is concerned with the development of algorithms that allow a computer to 'learn'. Given a set of data, a machine learning algorithm finds patterns or rules that characterise interesting aspects, or the structure, of the data. Constructing models of data observations is not a trivial task; in general, a model will only be an approximation to the true underlying data generating process. The problem lies in determining which aspects, or *features*, of the data are useful (in the way a human extracts useful sensory features in order to make sense of the environment) and capturing the way these features interact within the model.

In general, research in the machine learning field has focused on analysing data that is the output of a single sensor (a single data source) rather than analysing data from the output of several sensors. However, it seems advantageous to learn from multiple data sources because there is more information about the underlying data generating process than if we had just considered a single source. The relevance of this research area is inspired by the human brain's ability to integrate five different sensory input streams into a coherent representation of its environment. Additionally, due to the increased availability of electronic recording devices and advances in data analysis techniques, there exist many scenarios in which it becomes necessary to model multiple data sources. The analysis of more than one data source is of interest in fields such as robotics (where it is known as sensor fusion), data fusion of satellite observations, and multimodal image registration.

A naïve approach to the problem of multiple data set modelling would be to extract useful features for each data source in turn, and then combine the features together. Unfortunately, this approach neglects the potentially useful shared information between the data sources; since we suppose that the multiple sets of observations are views of the same underlying process, then the shared information will correspond to some knowledge about the process. In this thesis, we assume that the useful features of the data can be found through learning a joint representation of multiple data sets, and we create models that capture the interaction of these features. We can think of learning from each set as being guided by all the other sets i.e. the *context* guides the learning process for each data set. This suggests that inferring an underlying process from multiple sets of observations is more robust to error than learning from a single set of observations, since there is more information about the useful features.

1.1 Modelling two data sources

In this thesis, we focus on the case of two data sources, though the methods we consider may be generalised to multiple data sources. We suppose that we have N pairs of samples from the two data sources $\mathbf{Y}_1 = \{\mathbf{y}_{1,1}, ..., \mathbf{y}_{1,N}\}$ and $\mathbf{Y}_2 = \{\mathbf{y}_{2,1}, ..., \mathbf{y}_{2,N}\}$, where the nth pair is given by $\mathbf{y}_n = \{\mathbf{y}_{1,n}, \mathbf{y}_{2,n}\}$. There are many techniques in the literature for analysing two data sources, which we can categorise as either discriminative or generative models.

Discriminative techniques find pairs of features $\mathbf{x}_n = \{\mathbf{x}_{1,n}, \mathbf{x}_{2,n}\}$ for each data pair $\mathbf{y}_n$ to optimise a measure of similarity between the feature pair. The methods differ according to the relationship between $\mathbf{x}_n$ and $\mathbf{y}_n$, and the definition of the similarity measure. We can broadly categorise the different methods according to the relationship between the features and the data being linear, e.g. canonical correlation analysis (CCA) (Hotelling, 1936) or nonlinear e.g. kernel CCA (Lai & Fyfe, 2000). The nonlinear methods hold more interest than the linear methods, since in general most real life data sets can be described well by extracting nonlinearly related features. However, a difficult aspect of the modelling problem is the specification of the nonlinear mappings to find useful features; when finding nonlinear related features between two data sets, an overly flexible mapping may find spurious correlations between the data sets, and an inflexible mapping may not recover the true underlying relationship between the sets. In general, the difficulty with nonlinear problems is that there is an indeterminacy in the solution.

While there are many discriminative techniques for finding shared features between data sets, there are comparatively few generative techniques in the literature. The existing methods include probabilistic CCA (Bach & Jordan, 2005), a linear method, which formulates standard CCA as a Gaussian density estimation problem, and a nonlinear method in (Verbeek *et al.*, 2004), where each data set is modelled by a mixture of aligned local feature extractors. Generative methods for finding shared features are attractive because we can place a prior over the extracted features, and capture our intuition about the problem through the model structure. Additionally a joint density is defined over the two sets of data variables, allowing us to evaluate predictive densities of one data set given the other set.

1.2 Nonparametric methods

One of the problems of creating a model for a set of observed data is that in defining the model, strong assumptions are made about the underlying data generation process. If these assumptions are incorrect then the model will fail to capture the data's true underlying structure. Traditionally, the flexibility of the model stems from a set of model parameters, and training the model consists of finding the setting of the parameters which best fits the data. Suppose that we believe that a set of data points can be represented as coming from K distinct clusters. The problem lies in determining K.

Similarly, if we believe that the a set of data points are generated from an underlying function, then the problem lies in parameterising the function. If we choose a model structure that is too flexible, then the model will overfit the data. Conversely, if the model structure is too inflexible, the model will fail to find the underlying structure of the data.

Nonparametric Bayesian methods, originally developed in the statistics field, are rapidly receiving more interest in the machine learning community. Nonparametric Bayesian models have the attractive property that their complexity scales with the number of data points. The models that we create in this thesis are based on two types of nonparametric Bayesian models, Gaussian processes (O'Hagan, 1978; Williams & Rasmussen, 1996; Mackay, 1998; Rasmussen & Williams, 2006), which define a distribution over functions, and Dirichlet processes (Ferguson, 1973; Antoniak, 1974), which define a distribution over distributions.

1.3 Scope of the thesis

In this thesis, we focus on learning from two data sources. We use a generative probabilistic approach to the problem, such that each observation set consists of a shared component (which is conditionally independent on a shared latent variable) and models the between-set variation, and a non-shared component which models the within-set variation. We create three novel models which we discuss in Chapters 3, 4, and 5. The models are all nonparametric Bayesian methods, a field which has recently attracted a lot of interest in the machine learning community since this is an elegant way to define flexible models.

1.4 Overview of the thesis

Chapter 2

We introduce the problem of learning from two sets of observations, and discuss the advantages of a generative probabilistic approach over a discriminative approach. We then review a number of models in the literature for learning from data sources. The most important model that we discuss is probabilistic canonical correlation analysis (PCCA) (Bach & Jordan, 2005), which formulates canonical correlation analysis as a Gaussian density estimation problem. This is one of the few generative approaches to dependency seeking data analysis, and is a basis for the rest of the work in the thesis. We also introduce nonparametric Bayesian methods and discuss their use as flexible priors in probabilistic modelling.

Chapter 3

We examine the problem of learning from two sets of observations from an information theoretic perspective. We derive an alternative formulation of PCCA as probabilistic PCA (Tipping & Bishop, 1997) on two linearly transformed data sources, where the transformations are found automatically and capture the within-set variation in the data sources. We then extend this model, in the spirit of the Gaussian process latent variable model (GPLVM) (Lawrence, 2004) to create a GPLVM formulation of canonical correlation analysis. This is a generative probabilistic model of nonlinear canonical correlation analysis. We then evaluate GPLVM-CCA's performance on a range of data sets.

Chapter 4

We extend the model of the previous chapter to model complicated noise processes. Whereas the original model modelled the variance private to each data source as multivariate Gaussian, in this chapter we place Gaussian process priors on the noise function. The ability of the model to find shared and private components from two correlated data sources is demonstrated on synthetic data.

Chapter 5

We extend probabilistic canonical correlation analysis (PCCA) to a mixture of PCCA to model two data sources that lie close to nonlinear manifolds. We then further extend the model to a Dirichlet process mixture of PCCA, which allows the number of mixture components to be automatically determined from the data.

Chapter 6

Directions for future work are given in this chapter.

1.5 Publications

The thesis builds on work from the following publications:

- LEEN, G., & FYFE, C. 2006. A Gaussian Process Latent Variable Model Formulation of Canonical Correlation Analysis. *Pages 413–418 of: Proceedings of the 14th European Symposium of Artificial Neural Networks (ESANN)* (Chapter 3)

- FYFE, C., & LEEN, G. 2006. Stochastic Processes for Canonical Correlation Analysis. *Pages 245–50 of: Proceedings of the 14th European Symposium of Artificial Neural Networks (ESANN)* (Chapter 5)

1.6 Notation and conventions

In the mathematical notation, we use italics a to indicate scalars, bold lowercase $\mathbf{a}$ to indicate vectors, and bold uppercase $\mathbf{A}$ to indicate matrices. The vectors, unless otherwise stated, are column vectors. The transpose of a vector or a matrix is indicated by the superscript $\top$. The identity matrix is denoted by $\mathbf{I}$. Also, a subscript may be used to show $\mathbf{I}$'s dimension.

Chapter 2

Background

Given multiple sets of sensory data, an organism represents its knowledge about the world internally by means of synaptic structures in the brain; the internal representations are believed to be formed in such a way such that they are informative and can be used to reason about the environment. Incredibly, the outputs from the different senses are combined in such a way to create a coherent description of the world. This problem of jointly extracting the useful features from multiple different outputs is the focus of the thesis. The relevant information is extracted from each output in turn, depending on the current state of the other outputs, which we define as the context. We therefore refer to this type of learning as **context assisted learning**.

Learning from multiple sources of data sources is a timely problem. Due to the increasing availability of electronic recording devices, such as cameras and microphones, along with the advances in feature extraction of the recorded information from these sensors, there are many situations in which context assisted learning could be applied. Additionally, it is common to encounter multiple observations of the same phenomenon, yielding multiple sets of data which all share some common information. Some examples are:

- A human's five senses: sight, hearing, touch, taste, and smell, giving him /her five sets of observations of his/her environment.

- Many witnesses' accounts of an alleged crime.

- Translations of a set of documents in several different languages.

- Audio-visual person authentication.

Each example describes sets of observations of one phenomenon, such that there must be some shared information between the different observation sets. For instance, different translations of a document (as in the third example) will contain some shared information since the text in each of the translations will have the same meaning, regardless of language.

Suppose that we want to group a set of documents according to their topic. We want to learn a semantic (and language independent) representation of the text in the documents, which could be then be used for any retrieval or categorisation task in both a standard and cross-lingual scenario. The representation of the documents in this semantic, or topic, space is a compact way of expressing the information that is seen to be useful for this learning problem. By representing each document using the well known bag-of-words representation, i.e. as a vector of word count frequencies in a vector space where there is a dimension for every possible word in the vocabulary of the language, we would expect word occurrence patterns to indicate a particular topic. Across languages, these patterns will differ, but we would expect there to be correlations between the patterns for different translations of the same document. This problem was addressed in (Vinokourov *et al.*, 2003) by using a technique called Kernel Canonical Correlation Analysis.

Audio-visual person authentication systems attempt to verify the identity of a person through both an audio stream (such as the user speaking a sentence) and a corresponding video stream (of the user's face as he is speaking the sentence). Using two sources of information can yield better results than using only one; each stream can help the other to filter out the noise independent of the underlying process, and also to learn from incomplete data. For instance, parts of the video stream may be missing, due to noise or occlusion of the user's face. The audio stream can help to infer the missing parts of the video data, so that both can be used to jointly identify the user.

To summarise, the context assisted learning problem lies in combining the information from the multiple data sources so that we can find the most likely process underlying the observations. Since the sets of observations share a common source, it is expected that there are dependencies between the sets of observations. Learning consists of exploiting statistical regularities across multiple codings of the same process to extract common features. In this thesis, we derive several machine learning algorithms for modelling two data sources that share some common information. However, our methods may be generalised to modelling multiple related data sources.

2.1 A probabilistic view of the problem

For modelling two data sources, we want to find a compact representation of the information contained in both the data sources, in the same way a brain can compress two sources of information into an internal manageable representation. At first it may seem that a probabilistic approach is not necessary for the above problems. For instance, a possible solution to the problem of designing an audio-visual person authentication system is to construct a classifier which outputs a decision as to whether the identity claim is true. This involves finding a deterministic mapping from the audio-visual data to the decision, which does not involve any random variables. If we choose to output a measure of the uncertainty associated with the classifier, this requires estimation of the parameters of a binary random variable which does not call for the use of sophisticated probabilistic models. This type of approach is known as **discriminative modelling**, and is only concerned with optimising a mapping from the inputs to the desired outputs. By adjusting the classification boundary or function approximation accuracy, the model focuses on the given task to produce a good performance. Examples of discriminant models include support vector machines (Vapnik, 1995), and traditional neural networks e.g. (Haykin, 1994).

This approach neglects the true underlying structure to the problem; the generative process (such as the physical systems like the glottis and vocal tract that interact to create speech, the interaction of facial features to create the video signal) is not taken into account by the model. An approach that explicitly represents the underlying structure to a problem - the features (observed and unobserved) and probabilistic relationships between them - is called **generative modelling**. The generative approach defines a joint probability density over all the variables in the problem, which can then be manipulated to find desired classification or regression functions. Working in the joint distribution space offers a great degree of flexibility and a sense of completeness since we can insert knowledge about the system such as independencies, dependencies and prior distributions in a principled manner. For a comparison of discriminative and generative approaches, see (Jebara, 2001).

In this thesis, we choose to use generative models for finding dependencies between sets of observations. This is because the focus of generative modelling is to represent a phenomenon and resynthesise certain configurations from it, and for our problem we wish to represent the two data sources such that we can calculate quantities such as the predictive distribution over one source given the other, and the predictive distribution over the underlying processes given the data. Another reason for using generative models is that we are dealing with more than one set of observations, and probabilistic techniques are very good at reasoning in the increased complexity of the problem domain, due to the modelling of two sources instead of a single source. Furthermore, there are many existing generative models for finding an informative representation of a single data source, and the probabilistic framework allows these models to be extended in a principled way to the modelling of more than one data source.

The interdependencies between variables of a model can be represented simply through a **graphical model**, (also known as a directed acyclic graph or Bayesian network) and the structure and parameters of the model can be learned within the Bayesian framework from the data. Graphical models provide a good visual representation of the prior structure that we enforce on our generative model, which reflects our prior assumptions about the way in which the data is generated. Each node of the graphical model represents a random variable, and the arcs (or links) express the probabilistic relationships between the variables. We use directed graphical models, in which directed links (or arrows) are used to express conditional distributions. Unobserved random variables are denoted by shaded nodes in this thesis. A good introduction to graphical models is given in (Jordan, 1999) and (Frey, 1998).

2.2 Density estimation using parametric models

In this section, we review some techniques for fitting graphical models to data. Given a finite sample of data $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_N\}$, a common way of modelling this data is to assume that it is drawn from an unknown probability distribution $p(\mathbf{y})$, which we have to model. This method, called density estimation, allows us to summarise the data (see (Bishop, 1999) for an introduction). A standard approach to density estimation involves choosing a specific form for the density as a parametric model $p(\mathbf{y} \mid \Theta)$, which contains a number of adaptive parameters Θ. This approach is known as **parametric modelling**. Learning then consists of inferring the parameter values Θ given the observed data set $\mathbf{Y}$ i.e. finding $p(\Theta \mid \mathbf{Y})$. To infer the distribution over a data point $\mathbf{y}_n$ with the trained model, the following equation is used:

$$p(\mathbf{y}_n \mid \mathbf{Y}) = \int p(\mathbf{y}_n \mid \Theta)\, p(\Theta \mid \mathbf{Y})\, d\Theta \tag{2.1}$$

By integrating over Θ, we are considering all possible parameterisations of the model. The Bayesian approach is to estimate full distributions over the parameters, using Bayes rule, for use in (2.1):

$$p(\Theta \mid \mathbf{Y}) = \frac{p(\mathbf{Y} \mid \Theta)\, p(\Theta)}{p(\mathbf{Y})} = \frac{p(\mathbf{Y} \mid \Theta)\, p(\Theta)}{\int p(\mathbf{Y} \mid \Theta)\, p(\Theta)\, d\Theta} \tag{2.2}$$

In practice, it may be necessary to make approximations to (2.1), since computing the integral is not always straightforward. Two common approaches are **maximum likelihood** (ML) and **maximum a posteriori** (MAP) learning, which replace $p(\Theta \mid \mathbf{Y})$

with a point estimate Θ^* such that:

$$p(\mathbf{y} \mid \mathbf{Y}) = p(\mathbf{y} \mid \Theta^*) \text{ where } \Theta^* = \arg\max_{\Theta} p(\mathbf{Y} \mid \Theta) \quad \text{ML} \tag{2.3}$$

$$\Theta^* = \arg\max_{\Theta} p(\Theta \mid \mathbf{Y}) \quad \text{MAP} \tag{2.4}$$

In general, it is easier to maximise the log of the above probabilities, which results in the same solution. The maximum likelihood criterion only considers the training examples. The a posteriori estimate uses both the training examples and also a prior on Θ to regularise the estimate of Θ^*. When viewed as a function of the parameters Θ, $p(\mathbf{Y} \mid \Theta)$ is called the likelihood function. Choosing the likelihood function as an objective function for optimisation is intuitively appealing since if (as is often the case) the chosen model differs from the true distribution, maximisation of the likelihood corresponds to minimisation of the Kullback-Leibler divergence between the empirical distribution and the model. This results in the trained model approximating the empirical distribution subject to the constraints of modelling.

2.2.1 Latent variable models

A way of constraining the model is through the introduction of latent or hidden variables, which reduces the number of degrees of freedom in the model by expressing $p(\mathbf{y})$ in terms of a smaller number of variables. This makes the assumption that the intrinsic dimensionality of the data is lower than the data dimensionality i.e the data lies close to a manifold embedded in the data space. By fitting a generative latent variable model to the empirical data, it is expected that the latent variables capture some useful statistical properties about the observed data variables, and to reflect some aspect of the underlying data generating process. This lower dimensional latent representation of the observed variables can then be obtained by using Bayes rule. A latent variable model is defined by specifying the joint distribution over the latent variables $\mathbf{x} \in \Re^q$ and the observed variables $\mathbf{y} \in \Re^D$, where $q < D$. The joint distribution is decomposed as $p(\mathbf{y}, \mathbf{x}) = p(\mathbf{x})\, p(\mathbf{y} \mid \mathbf{x})$ where $p(\mathbf{x})$ is a prior distribution over the latent variables $\mathbf{x}$ and $p(\mathbf{y} \mid \mathbf{x})$ is a conditional distribution which expresses the uncertainty in the map-

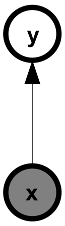


Figure 2.1: A graphical model for modelling a single data source $\mathbf{y}$ as generated by a latent (or hidden) variable $\mathbf{x}$.

ping from the latent variables to the observed variables. This structure is represented as the graphical model in Figure 2.1. The conditional distribution $p(\mathbf{y} \mid \mathbf{x})$ is expressed in terms of a mapping from $\mathbf{x}$ to $\mathbf{y}$. $\mathbf{y}$ is assumed to be generated from $\mathbf{x}$ according to:

$$\mathbf{y} = f(\mathbf{x}, \Theta) + \mathbf{n} \tag{2.5}$$

where $f(\mathbf{x}, \Theta)$ is a function of $\mathbf{x}$ parameterised by a set of parameters Θ, and $\mathbf{n}$ is a $\mathbf{x}$-independent zero mean noise process. After specifying the prior distribution $p(\mathbf{x})$, the desired distribution over $\mathbf{y}$ is found by marginalising out the latent variables:

$$p(\mathbf{y} \mid \Theta) = \int p(\mathbf{y} \mid \mathbf{x}, \Theta)\, p(\mathbf{x})\, d\mathbf{x} \tag{2.6}$$

Fitting the model to the data corresponds to determining the parameters Θ of the model by maximum likelihood, where the likelihood function is given by (2.6). Given N data samples $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_N\}$ and under the assumption that the samples are independently identically distributed (i.i.d) i.e. $p(\mathbf{Y} \mid \Theta) = \prod_{n=1}^{N} p(\mathbf{y}_n \mid \Theta)$, the log of the likelihood function is given by:

$$\mathcal{L} = \log p(\mathbf{Y} \mid \Theta) = \log \prod_{n=1}^{N} p(\mathbf{y}_n \mid \Theta) = \sum_{n=1}^{N} \log p(\mathbf{y}_n \mid \Theta) \tag{2.7}$$

In practice, the integral in (2.6) is intractable except for certain forms of $p(\mathbf{x})$ and $p(\mathbf{y} \mid \mathbf{x}, \Theta)$. One of the simplest latent variable models assumes that the observed variables are linearly related to the latent variables with added noise:

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \boldsymbol{\mu} + \mathbf{n} \tag{2.8}$$

where $\mathbf{y} \in \Re^D$, $\mathbf{x} \in \Re^q$, $\mathbf{W} \in \Re^{D\times q}$ is the matrix describing the linear relationship between $\mathbf{x}$ and $\mathbf{y}$, $\boldsymbol{\mu} \in \Re^D$ is a parameter vector allowing the model to have a non zero mean, and $\mathbf{n} \in \Re^D$ is a noise term, taken to be an independent sample from a Gaussian distribution with zero mean and covariance $\boldsymbol{\Sigma}_\mathbf{n}$:

$$p(\mathbf{n}) = N(\mathbf{n} \mid \mathbf{0}, \boldsymbol{\Sigma}_\mathbf{n})$$

This gives a Gaussian likelihood for a data point $\mathbf{y}_n$:

$$p(\mathbf{y}_n \mid \mathbf{x}_n, \mathbf{W}, \boldsymbol{\Sigma}_\mathbf{n}) = N(\mathbf{y}_n \mid \mathbf{W}\mathbf{x}_n + \boldsymbol{\mu}, \boldsymbol{\Sigma}_\mathbf{n}) \tag{2.9}$$

A conjugate prior is placed on the latent variables $p(\mathbf{x}_n) = N(\mathbf{x}_n \mid \mathbf{0}, \mathbf{I})$ and integrated out, giving a marginal likelihood:

$$\begin{aligned} p(\mathbf{y}_n \mid \mathbf{W}, \boldsymbol{\Sigma}_\mathbf{n}) &= \int p(\mathbf{y}_n \mid \mathbf{x}_n, \mathbf{W}, \boldsymbol{\Sigma}_\mathbf{n}) p(\mathbf{x}_n) d\mathbf{x}_n && (2.10) \\ &= N(\mathbf{y}_n \mid \boldsymbol{\mu}, \mathbf{W}\mathbf{W}^\top + \boldsymbol{\Sigma}_\mathbf{n}) && (2.11) \end{aligned}$$

The likelihood of the parameters given all N data points $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_N]^\top$ is given by

$$p(\mathbf{Y} \mid \mathbf{W}, \boldsymbol{\Sigma}_\mathbf{n}) = \prod_{n=1}^{N} N(\mathbf{y}_n \mid \mu, \mathbf{W}\mathbf{W}^\top + \boldsymbol{\Sigma}_\mathbf{n}) \tag{2.12}$$

(assuming that the data points are independent). Parameter values are then found to maximise the likelihood function (2.12). From inspection of (2.12), it can be seen that one solution would be $\mathbf{W} = 0$ and $\boldsymbol{\Sigma}_\mathbf{n} = \tilde{\boldsymbol{\Sigma}}$, the sample covariance matrix of the data. However, this is not an interesting solution since the data is solely modelled by noise; instead the form of $\boldsymbol{\Sigma}_\mathbf{n}$ is constrained such that $\mathbf{W}$ is forced to model interesting variation in the data e.g in factor analysis (Bartholomew, 1987), probabilistic principal component analysis (Tipping & Bishop, 1999), and probabilistic canonical correlation analysis (Bach & Jordan, 2005). For a more complete introduction to latent variable models, see (Bishop, 1999).

2.2.1.1 Finding the latent representation of the data

The latent representation $\mathbf{x}$ of the observed data $\mathbf{y}$ is found by applying Bayes rule. This can be thought of graphically as inverting the arrow of the graphical model in Figure 2.1. The posterior distribution over the latent variables is given by:

$$p\left(\mathbf{x} \mid \mathbf{y}, \Theta\right) = \frac{p\left(\mathbf{y}, \mathbf{x} \mid \Theta\right)}{p\left(\mathbf{y}\right)} \tag{2.13}$$

For the Gaussian model given in the previous section, it is known that since both $p\left(\mathbf{y}, \mathbf{x} \mid \Theta\right)$ and $p\left(\mathbf{y}\right)$ are Gaussian, the posterior density $p\left(\mathbf{x} \mid \mathbf{y}, \Theta\right)$ will also be Gaussian, with mean $\mu_{\mathbf{x}|\mathbf{y}}$ and covariance $\mathbf{\Sigma}_{\mathbf{x}|\mathbf{y}}$:

$$\begin{aligned} \mu_{\mathbf{x}|\mathbf{y}} &= \mathbf{W}^{\top}(\mathbf{W}\mathbf{W}^{\top} + \mathbf{\Sigma}_{\mathbf{n}})^{-1}(\mathbf{y} - \boldsymbol{\mu}) && (2.14) \\ \mathbf{\Sigma}_{\mathbf{x}|\mathbf{y}} &= \mathbf{I} - \mathbf{W}^{\top}(\mathbf{W}\mathbf{W}^{\top} + \mathbf{\Sigma}_{\mathbf{n}})^{-1}\mathbf{W} && (2.15) \end{aligned}$$

An equivalent formulation can be found by applying the Woodbury identity to the above equations giving:

$$\begin{aligned} \mu_{\mathbf{x}|\mathbf{y}} &= (\mathbf{W}^{\top}\mathbf{\Sigma}_{\mathbf{n}}^{-1}\mathbf{W} + \mathbf{I})^{-1}\mathbf{W}^{\top}\mathbf{\Sigma}_{\mathbf{n}}^{-1}(\mathbf{y} - \boldsymbol{\mu}) && (2.16) \\ \mathbf{\Sigma}_{\mathbf{x}|\mathbf{y}} &= (\mathbf{W}^{\top}\mathbf{\Sigma}_{\mathbf{n}}^{-1}\mathbf{W} + \mathbf{I})^{-1} && (2.17) \end{aligned}$$

The advantage of this formulation is that we only have to invert a $q \times q$ matrix rather than a $D \times D$ matrix in (2.14) and (2.15).

2.2.2 Extending latent variable models

We have reviewed a simple Gaussian latent variable model which can be used to model data that is thought to be linearly related to an underlying latent variable of lower dimensionality i.e. the data is assumed to lie close to a linear subspace. However, the model may not be sufficient for modelling more complex data sets, for instance the data may be better described as lying close to a nonlinear manifold. The linear latent variable model can be extended within the probabilistic framework to create more complex models that assume a nonlinear relationship between the latent and data spaces. One approach to this problem is to model the global nonlinear mapping. In this chapter we review two models which use this approach and can be viewed as probabilistic nonlinear principal component analysis models, the Generative Topographic Mapping (GTM) (Bishop *et al.*, 1998) and the Gaussian Process Latent Variable Model (GPLVM) (Lawrence, 2004). The second is to use a mixture of latent variable models as a set of local linear approximations to the nonlinear manifold, which is reviewed in Chapter 5.

2.3 Nonparametric Bayesian models

The models that we reviewed in the previous section are parametric models which assume some finite set of parameters Θ. Since the parameter set is finite, the complexity of the model is bounded, such that the model is not very flexible and may not be able to infer the correct model complexity for the data. Nonparametric models, on the other hand, assume an infinite set of parameters and hence are very flexible models.

2.3.1 Gaussian processes

Gaussian processes (GP) (O'Hagan, 1978; Williams & Rasmussen, 1996; Mackay, 1998; Rasmussen & Williams, 2006) are probability distributions over functions. In this section we illustrate how GP's can be used to infer the underlying function in a regression problem. Suppose that we have a supervised learning problem i.e. we want to learn a mapping from an input $\mathbf{x}$ to an output (or target) y from empirical data $\mathcal{D} = ((\mathbf{x}_i, y_i) \mid i = 1, .., N)$.

To make predictions of the target y given new input points $\mathbf{x}$, we need to find an underlying function f which will make predictions for all possible input values. We assume that the output is a noisy version of the function values $y = f(\mathbf{x}) + \mathbf{n}$ where $\mathbf{n}$

is i.i.d. Gaussian noise with variance β^{-1}. There are two main approaches to specifying the preferred characteristics of f. The first is to restrict the class of possible functions, such as in parametric modelling tools like the latent variable models introduced in Section 2.2.1, in which only linear functions of $\mathbf{x}$ are considered. The second approach specifies which functions are more preferable (for instance, functions that are smooth) by placing a prior over the space of all possible functions, giving higher probability to functions that have the desired characteristics. This second approach is more flexible since a rich class of functions can be considered. Gaussian process (GP) methods use this approach; a Gaussian process is the generalisation of a Gaussian probability distribution to a distribution over functions. Learning in the GP framework involves placing a prior over functions, then after seeing the data $\mathcal{D}$, calculating the posterior distribution over functions.

A formal definition for a Gaussian process is as follows. Consider a stochastic process which defines a distribution, $p(f)$, over functions, f, where f maps some input space, χ to $\Re$. If e.g. $\chi = \Re$, f is infinite dimensional; however the $\mathbf{x}$ values index the function, $f(\mathbf{x})$, at a countable number of points and so we use the data at these points to determine $p(f)$ in function space. If $p(f)$ is multivariate Gaussian for every finite subset of χ, the process is a GP and is then determined by a mean function $m(\mathbf{x})$ and covariance function $K(\mathbf{x}, \mathbf{x}')$:

$$m(\mathbf{x}) = E[f(\mathbf{x})] \tag{2.18}$$

$$K(\mathbf{x}, \mathbf{x}') = E[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \tag{2.19}$$

These are often defined by hyperparameters, expressing our prior beliefs on the nature of $K(\mathbf{x}, \mathbf{x}')$ and $m(\mathbf{x})$, whose values are learned from the data.

2.3.1.1 A regression example

Regression within the GP framework involves finding the underlying function f of the data y. We want to predict the function at a finite number of test input points which we denote by $\mathbf{X}^*$, given a training data set $\mathcal{D} = [\mathbf{X}, \mathbf{Y}]$. We first place a prior over the space of functions evaluated at $\mathbf{X}^*$; typically this a zero mean Gaussian process:

$$\mathbf{f}^* \sim N(\mathbf{f}^* \mid \mathbf{0}, K(\mathbf{X}^*, \mathbf{X}^*)) \tag{2.20}$$

To find the posterior distribution over functions (evaluated at $\mathbf{X}^*$) given the training data, $p(\mathbf{f}^* \mid \mathbf{y}, \mathbf{X}^*, \mathbf{X})$, we condition over the joint distribution
$p(\mathbf{f}^*, \mathbf{Y} \mid \mathbf{X}^*, \mathbf{X})$:

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{f}^* \end{pmatrix} \sim N\left(\begin{pmatrix} \mathbf{y} \\ \mathbf{f}^* \end{pmatrix} \mid \mathbf{0}, \begin{pmatrix} K(\mathbf{X}, \mathbf{X}) + \beta^{-1}\mathbf{I} & K(\mathbf{X}, \mathbf{X}^*) \\ K(\mathbf{X}^*, \mathbf{X}) & K(\mathbf{X}^*, \mathbf{X}^*) \end{pmatrix}\right) \tag{2.21}$$

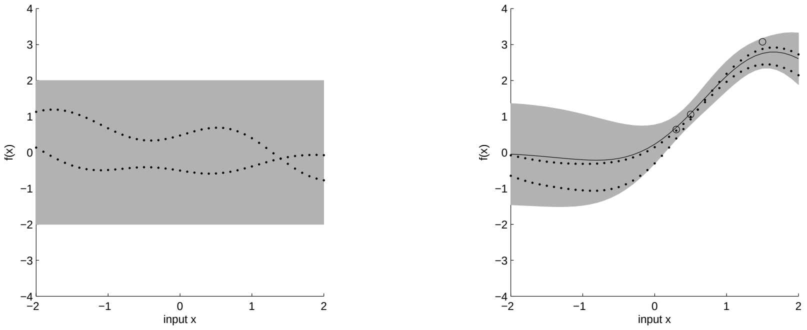
to gain

$$\begin{aligned} \mathbf{f}^* \mid \mathbf{X}^*, \mathbf{Y}, \mathbf{X} &\sim N(\mathbf{f}^* | \mu(\mathbf{X}^*), \sigma^2(\mathbf{X}^*)) && (2.22) \\ \text{where} & \\ \mu(\mathbf{X}^*) &= K(\mathbf{X}^*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \beta^{-1}\mathbf{I}]^{-1}\mathbf{y}, && (2.23) \\ \sigma^2(\mathbf{X}^*) &= K(\mathbf{X}^*, \mathbf{X}^*) - K(\mathbf{X}^*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \beta^{-1}\mathbf{I}]^{-1}K(\mathbf{X}, \mathbf{X}^*) && (2.24) \end{aligned}$$

Graphically we can think of inference in the GP framework as rejecting functions from the prior that do not agree with the observations $\mathcal{D}$. Figure 2.2 illustrates the inference steps for an example using 1-dimensional input and target variables.

2.3.2 Dirichlet processes

The Dirichlet process (DP) is a nonparametric distribution on distributions, or equivalently, a measure on measures (Ferguson, 1973). A DP is parameterised by a scaling parameter $\alpha_0 > 0$, and a base measure G_0. We can view DP's as an infinite dimensional Dirichlet distribution, which we review in the next section. The DP can be used as a nonparametric prior over the parameters of a mixture model (Ferguson, 1973; Antoniak, 1974; Escobar, 1994); in Chapter 5 we derive a DP mixture model of probabilistic canonical correlation analysers.



(a) (b)

Figure 2.2: Two functions $(\cdot)$ drawn at random from a GP prior, evaluated at $\mathbf{X}^*$ (a). Given the data set D and the prior, we can calculate the posterior distribution $\mathbf{f}^* \mid \mathbf{X}^*, \mathbf{Y}, \mathbf{X}$ over functions. (b) shows the data (o), the mean (-) of the posterior distribution (evaluated at $\mathbf{X}^*$), and two functions $(\cdot)$ drawn at random from the posterior distribution. In both diagrams, the grey shaded area represents the pointwise mean plus and minus 2 standard deviations for each input value for the prior and posterior respectively.

2.3.2.1 The Dirichlet distribution

The Dirichlet distribution is a distribution over discrete distributions (over the K dimensional probability simplex). Suppose that g is a K dimensional probability distribution on a discrete space, i.e. $\mathbf{g} = \{g_1, ..., g_K\}$ is a K dimensional vector s.t. $\forall i : g_i \geq 0$ and $\sum_{i=1}^K g_i = 1$. A Dirichlet distribution on g is written as:

$$p(\mathbf{g} \mid \boldsymbol{\alpha}') = Dir(\mathbf{g} \mid \alpha'_1, ..., \alpha'_K) = \frac{\Gamma(\sum_i \alpha'_i)}{\prod_i \Gamma(\alpha'_i)} \prod_{i=1}^K g_i^{\alpha'_i - 1} \tag{2.25}$$

where $\boldsymbol{\alpha}' = \{\alpha'_1, ..., \alpha'_K\}$ is the parameter vector and $\forall i : \alpha'_i > 0$. The first term is a normalisation constant, where $\Gamma(x) = \int_0^\infty u^{(x-1)} e^{-u} du$ denotes the Gamma function. The mean of the distribution is given by $\mathcal{E}(g_i) = \frac{\alpha'_i}{\sum_k \alpha'_k}$. This gives the probability that the probability of K events occurring are $\mathbf{g} = \{g_1, ..., g_K\}$, given that the ith event has been observed $\alpha'_i - 1$ times. It is convenient to reparameterise by defining:

$$\alpha_0 = \textstyle\sum_{i=1}^K \alpha'_i \quad \alpha_i = \frac{\alpha'_i}{\alpha_0}, i = 1, .., K \quad \boldsymbol{\alpha} = \{\alpha_1, ..., \alpha_K\} \tag{2.26}$$

With this formulation, $\mathcal{E}(g_i) = \alpha_i$, and α_0 can be considered as the *precision* or concentration parameter. When α_0 is large, g is likely to be near $\boldsymbol{\alpha}$, the mean of the distribution, and when α_0 is small, g can be spread far away around $\boldsymbol{\alpha}$.

2.3.2.2 Conjugacy to the multinomial distribution

The Dirichlet distribution is conjugate to the multinomial distribution. Suppose that we have a discrete observed variable Θ, having K possible states $\{\theta^1, ..., \theta^K\}$, such that $\Theta \sim$ Multinomial(g), with likelihood function:

$$p(\Theta = \theta^i \mid \mathbf{g}) = g_i, \text{ for } i = 1, ..., K \tag{2.27}$$

After observing $\Theta = \theta^i$, the posterior over g is also a Dirichlet:

$$p(\mathbf{g} \mid \Theta = \theta^i, \boldsymbol{\alpha}') = \frac{p(\mathbf{g} \mid \boldsymbol{\alpha}')p(\Theta = \theta^i \mid \mathbf{g})}{p(\Theta = \theta^i \mid \boldsymbol{\alpha}')} = Dir(\mathbf{g} \mid \boldsymbol{\alpha}'') \tag{2.28}$$

where $\boldsymbol{\alpha}'' = \{\alpha''_1, ..., \alpha''_K\}$ is the parameter vector. $\alpha''_i = \alpha'_i + 1$ and $\forall j \neq i : \alpha''_j = \alpha'_j$. This shows that the posterior over g is based on the updated 'counts' $\boldsymbol{\alpha}''$ of the observed states of Θ. For a data set $\mathcal{D} = \{\Theta_1, ..., \Theta_N\}$, ($N$ observed states of Θ), the posterior over g is:

$$p(\mathbf{g} \mid \mathcal{D}, \boldsymbol{\alpha}') = Dir(\mathbf{g} \mid \alpha'_1 + N_1, ..., \alpha'_K + N_K) \tag{2.29}$$

where N_i is the number of times $\Theta = \theta^i$ in $\mathcal{D}$. The probability of the next data point Θ_{N+1} given the observed data $\mathcal{D}$, is:

$$\begin{aligned} p(\Theta_{N+1} = \theta^i \mid \mathcal{D}, \boldsymbol{\alpha}') &= \int p(\Theta_{N+1} = \theta^i \mid \mathbf{g})p(\mathbf{g} \mid \mathcal{D}, \boldsymbol{\alpha}')d\mathbf{g} & (2.30) \\ &= \int g_i Dir(\mathbf{g} \mid \alpha'_1 + N_1, ..., \alpha'_K + N_K)d\mathbf{g} & (2.31) \\ &= \frac{\alpha_0\alpha_i + N_i}{\alpha_0 + N} & (2.32) \end{aligned}$$

This shows the effect of the Dirichlet prior over g, the parameters of the multinomial distribution. Without the prior, the maximum likelihood estimate of g is given by $g_i^{ML} = \frac{N_i}{N}, i = 1, ..., K$, which is a point estimate of $p(\mathbf{g} \mid \mathcal{D}, \boldsymbol{\alpha}')$. If some of the

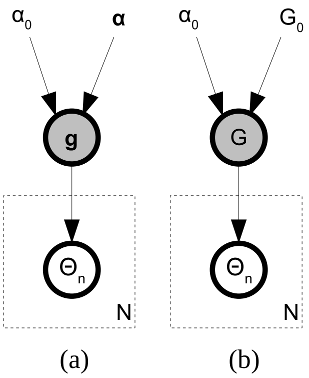


Figure 2.3: Graphical model for the Dirichlet distribution (a) and the Dirichlet process (b)

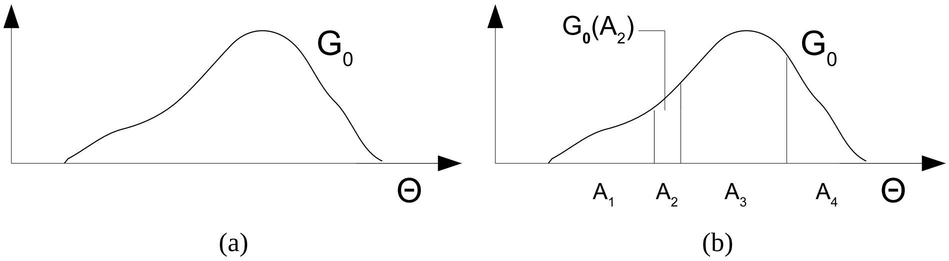


Figure 2.4: Illustration of a Dirichlet process prior on Θ

counts N_i are very small, and $N < K$, the parameters g may incorrectly be estimated to be zero. When using the Dirichlet prior as in (2.32), this tends towards the maximum likelihood estimate when the counts N_i become large and the data dominates the prior.

Figure 2.3a shows a generative model for $\mathcal{D} = \{\Theta_1, ..., \Theta_N\}$. This combines a multinomial likelihood model with a Dirichlet prior; a distribution over Θ_n is generated from the Dirichlet prior $p(\text{g} \mid \alpha_0, \boldsymbol{\alpha})$, and then a value for Θ_n is drawn from $\Theta_n \sim$ Multinomial(g). It is not straightforward to sample from g; an alternative is to sample Θ_n by directly (integrating over g) using the predictive distribution in (2.32), where $\mathcal{D}$ is the previously generated samples.

Suppose that G_0 is a distribution over a measurable space Θ, as depicted in Figure 2.4a. This acts as the base measure for the DP, and this can be interpreted as the continuous version of the parameter vector $\boldsymbol{\alpha} = \{\alpha_1, ..., \alpha_K\}$, the mean of the Dirichlet distribution. A Dirichlet process is defined to be the distribution of a random probability measure G over Θ i.e.

$$G \sim DP(G \mid G_0, \alpha_0) \tag{2.33}$$

such that for any K finite partitions of Θ, $\{A_1, ..., A_K\}$, (as shown in Figure 2.4b), $\{G(A_1), ..., G(A_K)\}$ follows a finite dimensional Dirichlet distribution with parameters $\{\alpha_0 G_0(A_1), ..., \alpha_0 G_0(A_K)\}$:

$$\{G(A_1), ..., G(A_K)\} \sim Dir(\alpha_0 G_0(A_1), ..., \alpha_0 G_0(A_K)) \tag{2.34}$$

where $\alpha_0 > 0$ determines the concentration of $\{G(A_1), ..., G(A_K)\}$ around $\{G_0(A_1), ..., G_0(A_K)\}$. As $\alpha_0 \to \infty$, $G \to G_0$. The graphical model for the Dirichlet process is shown in Figure 2.3b.

The posterior over G given $\mathcal{D} = \{\Theta_1, ..., \Theta_N\}$ is given by:

$$p(G \mid \mathcal{D}, \alpha_0, G_0) = DP\left(G \middle| \frac{1}{\alpha_0 + N}\left(\alpha_0 G_0 + \sum_{i=1}^{N} \delta_{\Theta_i}\right), \alpha_0 + N\right) \tag{2.35}$$

where δ_{Θ_i} is a discrete measure (or atom) concentrated at Θ_i. The Dirichlet process allows us to model deviations away from a baseline prior G_0. We present two perspectives on the Dirichlet process.

2.3.2.3 Pólya Urn Scheme

One perspective on the Dirichlet process is provided by the Pólya urn scheme (Blackwell & MacQueen, 1973), which demonstrates the clustering property of draws from G. Suppose that we have already generated a sequence of N data points $\mathcal{D} = \{\Theta_1, ..., \Theta_N\}$ according to G; $\{\Theta_1, ..., \Theta_N\}$ are conditionally independent given G, and exchangeable. Integrating over G, we get

$$\begin{aligned} p(\Theta_{N+1} \mid \mathcal{D}, \alpha_0, G_0) &= \int p(\Theta_{N+1} \mid G) p(G \mid \mathcal{D}, \alpha_0, G_0) dG && (2.36) \\ &= \frac{1}{\alpha_0 + N}\left(\alpha_0 G_0(\Theta_{N+1}) + \sum_{i=1}^{N} \delta_{\Theta_i}\right) && (2.37) \end{aligned}$$

With the Pólya urn sampling scheme, we assume that there is an urn which contains coloured balls. The balls are drawn from the urn with probability proportional to their mass. The coloured balls have unit mass and there is an additional black ball that has mass α_0. After drawing a coloured ball from the urn, we replace the ball in the urn with *an additional ball of the same colour*. If the black ball is drawn, it is replaced along with a ball of a new colour, where the colour is drawn from distribution G_0. In (2.37), a data point Θ_n represents a draw from the urn, $\mathcal{D} = \{\Theta_1, ..., \Theta_N\}$ is the current state of the urn, and the ith colour is represented by θ_i. If we rewrite (2.37) as:

$$p(\Theta_{N+1} \mid \mathcal{D}, \alpha_0, G_0) = \frac{\alpha_0}{\alpha_0 + N} G_0(\Theta_{N+1}) + \frac{N}{\alpha_0 + N} \left(\frac{1}{N} \sum_{i=1}^{N} \delta_{\Theta_i} \right) \tag{2.38}$$

we can see that it is a mixture of distributions. With probability $\frac{\alpha_0}{\alpha_0+N}$, Θ_{N+1} is drawn from G_0, as we can see from the first term of (2.38). Analogously, this is the probability that we draw the black ball from the urn. The second term of (2.38) shows that with probability $\frac{N}{\alpha_0+N}$, Θ_{N+1} is drawn uniformly from $\{\Theta_1, ..., \Theta_N\}$, or equivalently, one of the coloured balls is drawn from the urn (and hence the new ball takes on the same colour as one of the existing balls). The values of the previous data points (or balls in the urn) are not necessarily distinct. The probability that $\Theta_n = \theta_i$ (is the ith colour) is given by:

$$p(\Theta_{N+1} = \theta^i \mid \mathcal{D}, \alpha_0, G_0) = \frac{N_i}{\alpha_0 + N} \tag{2.39}$$

The Pólya urn sampling scheme shows the clustering property of the draws from G, in that a set of samples $\{\Theta_1, ..., \Theta_N\}$ are not necessarily distinct. This means that the data is divided into K partitions, or clusters, where each partition has the same parameter setting θ^i. The more often θ^i is drawn, the more likely it is to be drawn in the future. α_0 controls the tendency to form clusters; if α_0 is very small, it is likely that there will be few clusters, and if α_0 is large, there will be many small clusters. Another analogy for the clustering mechanism is given by the Chinese restaurant process.

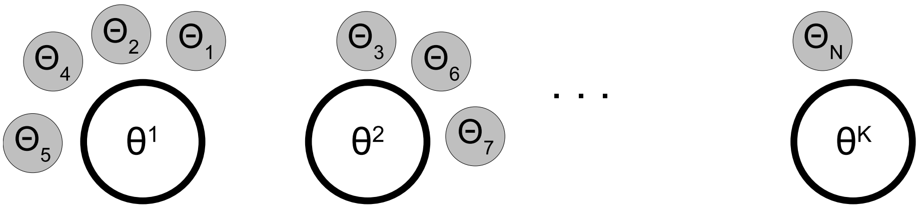


Figure 2.5: The Chinese restaurant process. The customers (Θ_n) are seated at the tables (circles), where the kth table corresponds to the unique value θ^k.

2.3.2.4 Chinese restaurant process

In the Chinese restaurant process (Aldous, 1985), N customers sit down in the restaurant which has an infinite number of tables. The tables represent the distinct values $\theta^i, i = 1, ..., K$, where K is the number of occupied tables, or *represented clusters*. The ith customer represents Θ_i.

- The first customer Θ_1 sits at the first table θ^1. $N_1 = 1$, $K = 1$.

- Either the ith customer sits at already occupied table θ^k with probability

$$\frac{N_{-i,k}}{\alpha_0 + N} \tag{2.40}$$

  where $N_{-i,k}$ denotes the number of customers at table k, not including the current customer. The ith customer inherits θ^k. $N_k \leftarrow N_k + 1$.

- or with probability

$$\frac{\alpha_0}{\alpha_0 + N} \tag{2.41}$$

  the ith customer sits at a new table, θ^{k+1}. For the new table, θ^{k+1} is generated from G_0. $N_{K+1} = 1$, $K \leftarrow K + 1$.

The Chinese restaurant process is shown in Figure 2.5. The data points (the customers) Θ_n are clustered according to the parameter θ^k they have inherited (the table which they are occupying).

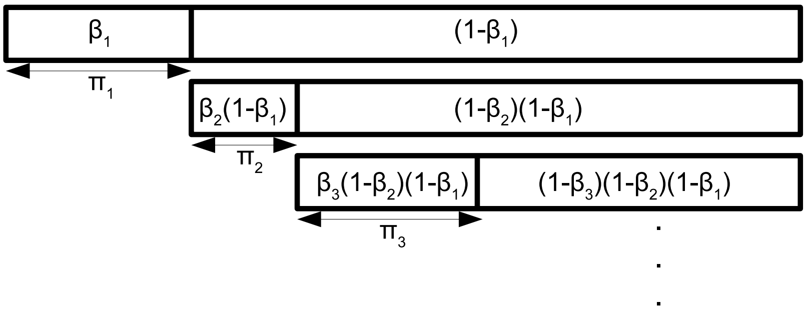


Figure 2.6: The stick breaking construction

2.3.2.5 Stick breaking representation

We can get an insight into G, the distribution drawn from a $DP(G \mid G_0, \alpha_0)$, through the stick breaking construction (Sethuraman, 1994). G can be represented as:

$$G = \sum_{i=1}^{\infty} \pi_i \delta_{\theta^i} \tag{2.42}$$

where δ_{θ^i} is a probability measure concentrated at θ^i, and π_i and θ^i are defined below. The stick breaking construction is based on two independent infinite sequences of independent random variables $\{\beta_i\}_{i=1}^{\infty}$ and $\{\theta^i\}_{i=1}^{\infty}$:

$$\beta_i \sim Beta(1, \alpha_0) \tag{2.43}$$

$$\theta^i \sim G_0 \tag{2.44}$$

The infinite sequence $\boldsymbol{\pi} = \{\pi_i\}_{i=1}^{\infty}$ is defined recursively as:

$$\pi_i = \beta_i \prod_{j=1}^{i-1} (1 - \beta_j) \tag{2.45}$$

which can be interpreted as breaking of parts of a stick, initially of unit length, as depicted in Figure 2.6, and therefore we write $\boldsymbol{\pi} \sim \text{Stick}(\alpha_0)$. We can show that $\sum_{i=1}^{\infty} \pi_i = 1$ since $1 - \sum_{i=1}^{K} \pi_i = \prod_{i=1}^{K} (1 - \beta_i) \xrightarrow{K \to \infty} 0$. This shows that $\boldsymbol{\pi}$ can be interpreted as a random probability measure on positive integers.

2.4 Modelling a single data source

In this section, we look at different ways of modelling a single data source, since later on we extend these models to modelling more than one data source.

2.4.1 Probabilistic principal component analysis

Principal component analysis can be obtained from a specific form of latent variable model, as will be seen in this section. Principal component analysis (PCA) (Joliffe, 1986) is a well established statistical technique for dimensionality reduction. In general, mapping the data into a lower dimensional space is accompanied by the loss of some information contained in the data, so a desired property of a dimensionality reduction technique is to preserve as much of the useful information as possible. Given a set of N D-dimensional data vectors $\mathbf{y}_n, n \in \{1, ..., N\}$, the principal axes $\mathbf{u}_j, j \in \{1, ..., D\}$, are defined as the eigenvectors of the sample covariance matrix $\tilde{\mathbf{\Sigma}} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{y}_n - \mu_{\mathbf{y}})(\mathbf{y}_n - \mu_{\mathbf{y}})^\top$, where $\mu_{\mathbf{y}}$ is the sample mean of the data, such that

$$\tilde{\mathbf{\Sigma}}\mathbf{U} = \mathbf{U}\mathbf{\Lambda} \tag{2.46}$$

where $\mathbf{U}$ is the matrix of column eigenvectors $\mathbf{u}_j, j \in \{1, ..., D\}$, and $\mathbf{\Lambda}$ is the diagonal matrix of corresponding eigenvalues $\lambda_j, j \in \{1, ..., D\}$. The principal components are given by the linear projection of the data onto the principal axes. For a data point $\mathbf{y}_n$, the principal components are given by $\mathbf{x}_n = \mathbf{U}^\top\mathbf{y}_n$. Suppose that we only retain a subset $q < D$ of the principal axes, i.e. the q dominant eigenvectors of $\tilde{\mathbf{\Sigma}}$, as the columns of the matrix $\mathbf{U}_q \in \Re^{D\times q}$. By projecting the data onto $\mathbf{U}_q$, a reduced dimensionality representation (q-dimensional) of the data is obtained. For the n-th data point $\mathbf{y}_n$ the corresponding latent variable is given by $\mathbf{x}_n = \mathbf{U}_q^\top\mathbf{y}_n$. These projections are of interest because they minimise the squared reconstruction error over the whole data set $\mathbf{Y}$:

$$E_q = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{y}_n - \tilde{\mathbf{y}}_n)^2 \tag{2.47}$$

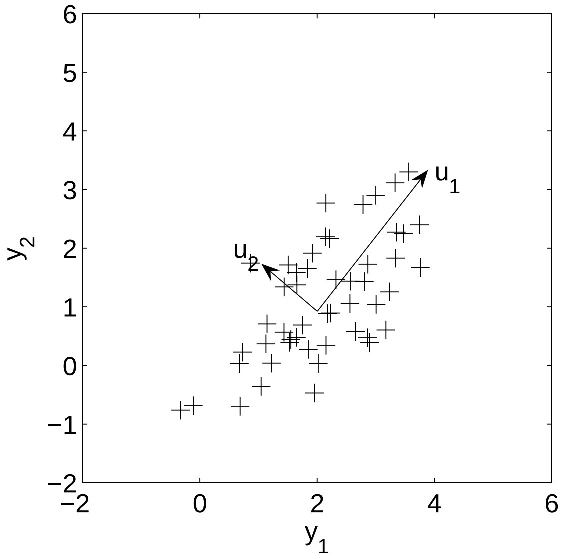


Figure 2.7: Illustration of principal component analysis applied to two dimensional data.

where $\tilde{\mathbf{y}}_n = \mathbf{U}_q \mathbf{x}_n$ is the reconstruction of the nth data point. This formulation of PCA suggests an alternative approach to finding the principal components of the data, by minimising (2.47). This approach forms the basis for nonlinear extensions of PCA.

A probabilistic formulation of PCA called probabilistic PCA (PPCA) was introduced in (Tipping & Bishop, 1999) in the form of a Gaussian latent variable model. By assuming that the noise covariance is isotropic, i.e. $\mathbf{\Sigma_n} = \sigma^2 \mathbf{I}$, PCA can be derived from within a Gaussian density estimation framework as in Section 2.2.1. For this noise model, the log likelihood is given by:

$$\begin{aligned} L &= \sum_{n=1}^{N} \log p(\mathbf{y}_n) \\ &= -\frac{ND}{2} \log 2\pi - \frac{N}{2} \log |\mathbf{C}| - \frac{N}{2} \mathrm{Tr}\{\mathbf{C}^{-1} \tilde{\mathbf{\Sigma}}\} \end{aligned} \tag{2.48}$$

where $\mathbf{C} = \mathbf{W}\mathbf{W}^\top + \sigma^2 \mathbf{I}$ and $\tilde{\mathbf{\Sigma}}$ is the sample covariance matrix of the data. There exists an exact analytical solution for the parameters of the model $\mathbf{W}$ and σ^2; the maximum likelihood solution of the parameters $\mathbf{W}_{ML}$ and σ^2_{ML} (obtained by maximising

(2.48) with respect to $\mathbf{W}$ and σ^2) are given by:

$$\mathbf{W}_{ML} = \mathbf{U}_q(\mathbf{\Lambda}_q - \sigma^2_{ML}\mathbf{I})^{\frac{1}{2}}\mathbf{R} \tag{2.49}$$

$$\sigma^2_{ML} = \frac{1}{D-q}\sum_{j=q+1}^{D}\lambda_j \tag{2.50}$$

where $\mathbf{U}_q \in \Re^{D\times q}$ is a matrix whose columns are the first q eigenvectors of $\tilde{\Sigma}$ i.e. the first q principal axes, with corresponding eigenvalues $\lambda_j, j = 1, ..., q$ in the diagonal matrix $\mathbf{\Lambda}_q \in \Re^{q\times q}$, and $\mathbf{R}$ is a rotation matrix. Suppose that we now want to find the latent variable representation of the data. This is found by evaluating the posterior density over the latent variables. Using (2.16) and (2.17) and the ML estimates for the parameters, we get:

$$\begin{aligned} p(\mathbf{x}_n \mid \mathbf{y}_n) &= N(\mathbf{x}_n \mid \mu_{\mathbf{x}|\mathbf{y}}, \Sigma_{\mathbf{x}|\mathbf{y}}) && (2.51)\\ \text{where } \mu_{\mathbf{x}|\mathbf{y}} &= (\mathbf{W}^\top_{ML}\mathbf{W}_{ML} + \sigma^2_{ML}\mathbf{I})^{-1}\mathbf{W}^\top_{ML}\mathbf{y}_n && (2.52)\\ \Sigma_{\mathbf{x}|\mathbf{y}} &= \sigma^2_{ML}(\mathbf{W}^\top_{ML}\mathbf{W}_{ML} + \sigma^2_{ML}\mathbf{I})^{-1} && (2.53) \end{aligned}$$

The reduced dimensionality representation for a data point $\mathbf{y}_n$ can be obtained by summarising $p(\mathbf{x}_n \mid \mathbf{y}_n)$ by its mean, which is given in (2.52). Due to the noise variance σ^2_{ML} this does not represent an orthogonal projection into latent space as in standard PCA, since the latent projection becomes skewed towards the origin. If we let $\sigma^2_{ML} \to 0$ when defining the model, the density model will become singular and therefore undefined. However, if necessary we can still obtain the optimal reconstruction of the data from the latent mean by omitting the noise term in the reconstruction by using $\tilde{\mathbf{y}}_n = (\mathbf{W}^\top_{ML}\mathbf{W}_{ML})^{-1}\mathbf{W}^\top_{ML}\mathbf{y}_n$.

2.4.2 Nonlinear PCA

The PPCA model is limited since we only find latent representations that are linearly related to the data, and we can only model the data as coming from a unimodal Gaussian density. In this section we consider a latent space that is nonlinearly related to the data space. Nonlinear dimensionality reduction is generally an ill posed problem, since the space of nonlinear functions is very large and hence there will not be a unique solution when fitting a nonlinear function to data. To overcome this problem, we have to constrain the form of the solution, as we will see in the following sections.

Suppose that we have a data set that is intrinsically low dimensional but is embedded nonlinearly in a high dimensional space i.e. it lies on, or close to, a nonlinear manifold. This is a generalisation of the linear dimensionality reduction problems that we reviewed in Section 2.2.1, but whereas before we restricted our analysis to finding linear transformations of the data i.e. approximation of the data by a linear subspace, we now consider any nonlinear mapping, giving us a nonlinear principal component analysis problem. In this context, looking for the greatest nonlinear direction of variance in the data is problematic. Instead, nonlinear PCA type methods try to find a manifold which minimises the squared reconstruction error.

One approach to constructing a nonlinear model is to assume that linear approximations can be made in local regions of the data space. In (Tipping & Bishop, 1997), the authors extend their probabilistic model of PCA to create a well defined mixture model of principal component analysers, whose parameters can be estimated by an EM algorithm, to capture data that lies on a nonlinear manifold. In this method, the nonlinear manifold is approximated by linear PCA models. A nonlinear latent variable model called the generative topographic mapping (GTM) was introduced in (Bishop *et al.*, 1996), (see also (Svensén, 1998; Bishop *et al.*, 1998)) where the nonlinear function $f(\mathbf{x}, \Theta)$ of the latent variable $\mathbf{x} \in \Re^q$ underlying the data is given by a generalised linear regression model of the form:

$$f(\mathbf{x}, \Theta) = \mathbf{W}\phi(\mathbf{x}) \tag{2.54}$$

where $\mathbf{W} \in \Re^{D \times M}$, and $\phi(\mathbf{x}) \in \Re^M$, whose elements $\phi_j(\mathbf{x})$ consist of M fixed basis functions evaluated at $\mathbf{x}$. The relationship between the latent and data variables is given by the mapping with some added noise $\mathbf{n} \in \Re^D$ which is taken to be from an isotropic

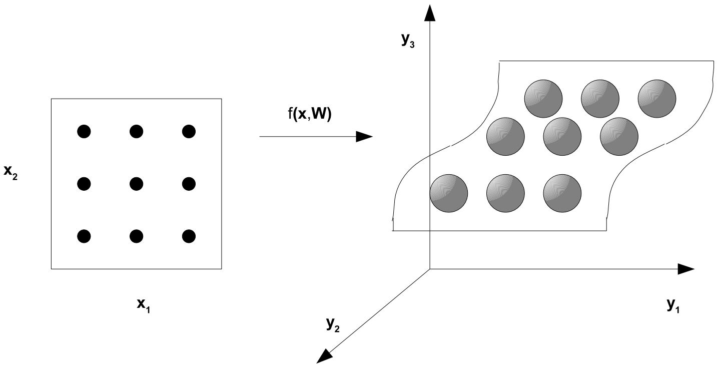


Figure 2.8: Schematic illustration of the GTM: a grid of latent points is mapped through a parameterised nonlinear mapping $f(\mathbf{x}, \mathbf{W})$ to a corresponding grid of Gaussian centres embedded in data space. Adapted from (Bishop *et al.*, 1996)

Gaussian distribution with variance σ^2. The conditional distribution $p(\mathbf{y}|\mathbf{x}, \mathbf{W}, \sigma^2)$ is given by:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{W}, \sigma^2) = N(\mathbf{y} \mid \mathbf{W}\phi(\mathbf{x}) + \boldsymbol{\mu}, \sigma^2\mathbf{I}) \tag{2.55}$$

where $\boldsymbol{\mu}$ is typically incorporated as a bias term into the basis functions. As we mentioned in Section 2.2.1, the integral in (2.6) is generally intractable; in order to formulate a tractable nonlinear latent variable model the prior distribution is chosen to be:

$$p(\mathbf{x}) = \frac{1}{K}\sum_{k=1}^{K}\delta(\mathbf{x} - \mathbf{x}_k) \tag{2.56}$$

i.e. a set of K equally weighted delta functions on a regular grid. The integral in (2.6) becomes a sum:

$$p(\mathbf{y}|\mathbf{W}, \sigma^2) = \frac{1}{K}\sum_{k=1}^{K}p(\mathbf{y} \mid \mathbf{x}_k, \mathbf{W}, \sigma^2) \tag{2.57}$$

Each delta function maps to the centre of an isotropic Gaussian which lies on a manifold nonlinearly embedded in data space. If $f(\mathbf{x}, \mathbf{W})$ is chosen to be continuous, then the ordering of the centering of the Gaussians in data space corresponds to the ordering of the latent points, as shown in Figure 2.8, i.e. the topography of the data is preserved in its latent representation. Since the centres of the Gaussians cannot move independently of each other, since they are constrained by the mapping $f(\mathbf{x}, \mathbf{W})$, the GTM can be viewed as a constrained mixture of Gaussians.

To train the model, the log likelihood function is maximised, which could be achieved by any standard nonlinear optimisation technique, but the authors use the Expectation Maximisation algorithm (Dempster *et al.*, 1977) due to the model's similarity to a mixture of Gaussians.

One of the disadvantages of the nonlinear mapping associated with the GTM, as noted by its authors, is due to the parameterisation. It requires a decision on the number of fixed basis functions M, which puts a hard constraint on the mapping's flexibility. Rather than using a generalised regression model, a Gaussian process can be used instead which allows the flexibility of the nonlinear mapping to be determined by the hyperparameters of the covariance function.

2.4.3 The GPLVM

The Gaussian Process Latent Variable Model (GPLVM) was introduced in (Lawrence, 2004, 2005). Latent variable models are parametric models; they assume a certain form for the data density and thus may be a bad fit for the data if the true density is very different to the model's assumptions. A novel interpretation of Probabilistic PCA, termed Dual Probabilistic Principal Component Analysis (DPPCA) takes the alternative approach of marginalising the parameters and optimising the latent variables. For a particular choice of Gaussian likelihood and prior, DPPCA turns out to be equivalent to the standard PPCA model, and a special case of a more general class of models, Gaussian Process Latent Variable Models (GPLVM). The GPLVM can be viewed as a nonparametric model since the mapping between the latent and data space is not explicitly parameterised.

The GPLVM uses Gaussian Processes (GP's) in an unsupervised manner for nonlinear dimensionality reduction. The inputs to the GP's or latent variables are mapped to a distribution over the data space by D independent GP's, where D is the dimensionality of the data space. The latent coordinates and the hyperparameters of the GP covariance function are then adjusted to maximise the GP likelihood. To show the link between latent variable models and Gaussian processes, we now study the DPPCA model. In (Lawrence, 2004), a conjugate prior is placed on the linear mapping $\mathbf{W}$ of the PPCA model, $p(\mathbf{W}) = \prod_{i=1}^{D} N(\mathbf{w}_i \mid \mathbf{0}, \mathbf{I})$, where $\mathbf{w}_i$ is the ith row of $\mathbf{W}$, and then $\mathbf{W}$ is marginalised giving a likelihood:

$$\begin{aligned} p(\mathbf{Y} \mid \mathbf{X}) &= \prod_{n=1}^{N} \int p(\mathbf{y}_n \mid \mathbf{x}_n, \mathbf{W}, \beta) p(\mathbf{W}) d\mathbf{W} && (2.58) \\ &= \prod_{d=1}^{D} N(\mathbf{Y}_{:,d} \mid 0, \mathbf{X}\mathbf{X}^\top + \beta^{-1}\mathbf{I}) && (2.59) \\ &= \frac{1}{(2\pi)^{\frac{DN}{2}} |\mathbf{K}|^{\frac{D}{2}}} \text{exp}(-\frac{1}{2}\text{tr}(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^\top)) && (2.60) \end{aligned}$$

where we have used (2.9) and the PPCA noise model from Section 2.4.1 in (2.58) , $\mathbf{\Sigma}_n = \beta^{-1}\mathbf{I}$, where β is the inverse noise variance, $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_N]^\top$ with corresponding latent variables $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]^\top$, $\mathbf{Y}_{:,i}$ denotes the ith column of $\mathbf{Y}$ i.e. the N independent realisations of the ith data dimension, and $\mathbf{K} = \mathbf{X}\mathbf{X}^\top + \beta^{-1}\mathbf{I}$. The log likelihood is given by the log of (2.60):

$$L = -\frac{DN}{2}\text{ln}(2\pi) - \frac{D}{2}\text{ln}|\mathbf{K}| - \frac{1}{2}\text{tr}(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^\top) \tag{2.61}$$

Writing $\mathbf{S} = D^{-1}\mathbf{Y}\mathbf{Y}^\top$, we optimise the log likelihood with respect to $\mathbf{X}$, giving

$$\frac{\partial L}{\partial \mathbf{X}} = -\mathbf{K}^{-1}\mathbf{S}\mathbf{K}^{-1}\mathbf{X} + \mathbf{K}^{-1}\mathbf{X} = 0 \tag{2.62}$$

Pre-multiplying by $\mathbf{K}$ gives

$$\mathbf{S}[\beta^{-1}\mathbf{I} + \mathbf{X}\mathbf{X}^\top]^{-1}\mathbf{X} = \mathbf{X} \tag{2.63}$$

Substituting $\mathbf{X}$ with its eigendecomposition, $\mathbf{X} = \mathbf{ULR}^\top$ gives

$$\begin{aligned} \mathbf{S}(\mathbf{U}[\beta^{-1}\mathbf{I} + \mathbf{L}^2]^{-1}\mathbf{U}^\top)\mathbf{ULR}^\top &= \mathbf{ULR}^\top \\ \mathbf{SU}[\mathbf{L} + \beta^{-1}\mathbf{L}^{-1}]^{-1}\mathbf{R}^\top &= \mathbf{ULR}^\top \end{aligned} \tag{2.64}$$

Right multiplying both sides by $\mathbf{R}$, we get

$$\mathbf{SU} = \mathbf{U}(\beta^{-1}\mathbf{I} + \mathbf{L}^2) \tag{2.65}$$

so that $\mathbf{U}$ are eigenvectors of $\mathbf{S}$ with eigenvalues $(\beta^{-1}\mathbf{I} + \mathbf{L}^2)$, giving

$$\mathbf{X} = \mathbf{U}_q\mathbf{LR}^\top \tag{2.66}$$

where $\mathbf{U}_q \in \Re^{N\times q}$ is a matrix whose columns are the first q eigenvectors of $\mathbf{YY}^\top$, $\mathbf{L}$ is a diagonal matrix whose jth element is $l_j = (\frac{\lambda_j}{D} - \beta^{-1})^{\frac{1}{2}}$, where λ_j is the eigenvalue associated with the qth eigenvector of $\mathbf{YY}^\top$, and $\mathbf{R}$ is a rotation matrix.

This eigenvalue problem is equivalent to that solved in PCA; $\mathbf{X}$ are the projections of the data onto the principal component axes, and DPPCA has the same underlying structure as PPCA. We note that the DPPCA model has the advantage in that it can easily be extended to allow for nonlinear processes by replacing the inner product kernel K with a nonlinear covariance function. (Lawrence, 2004) refers to this general class of models as Gaussian Process Latent Variable Models, due to the Gaussian process ‘mappings’ from the latent space to distributions over the data space.

2.4.4 Kernel Principal Component Analysis

Kernel methods are a relatively new family of algorithms that combine the simplicity of linear algorithms with the flexibility of nonlinear systems. The basis of kernel methods is to embed the data into a Hilbert space and to find linear relations within this space. The embedding of the data in this space is performed implicitly - the embeddings are defined in terms of inner products between pairs of points in the new space rather than explicitly by their coordinates. This is known as 'the kernel trick'. Therefore, kernel methods can be viewed as a way of nonlinearising linear algorithms that depend only on inner products between data points.

Suppose that we have a data space (or input space) $\mathcal{Y}$ and an embedding vector space (or feature space) $\mathcal{F}$ and we define a feature map $\phi : \mathcal{Y} \rightarrow \mathcal{F}$. Given two data points $\mathbf{y}_i \in \mathcal{Y}$ and $\mathbf{y}_j \in \mathcal{Y}$, the corresponding feature vectors $\phi(\mathbf{y}_i)$ and $\phi(\mathbf{y}_j)$ are not calculated explicitly, but instead, their inner product is defined by the kernel function $k(\mathbf{y}_i, \mathbf{y}_j) = \phi(\mathbf{y}_i)^\top \phi(\mathbf{y}_j)$. Principal component analysis, as we reviewed in Section 2.4.1, is conventionally defined in terms of the covariance, or outer product matrix of the data $\mathbf{Y} = [\mathbf{y}_1^\top, ..., \mathbf{y}_N^\top]^\top$ (which we have assumed to be zero mean), $\tilde{\boldsymbol{\Sigma}}_y = \frac{1}{N} \sum_{n=1}^{N} \mathbf{y}_n \mathbf{y}_n^\top = \frac{1}{N} \mathbf{Y}^\top \mathbf{Y}$. This is called the primal formulation of the problem. To derive the dual formulation, it is noted that the principal axes $\mathbf{U}$ lie in the span of $\mathbf{Y}$ since:

$$\mathbf{U} = \tilde{\boldsymbol{\Sigma}}_y \mathbf{U} \boldsymbol{\Lambda}^{-1} = \frac{1}{N} \mathbf{Y}^\top (\mathbf{Y} \mathbf{U} \boldsymbol{\Lambda}^{-1}) \tag{2.67}$$

i.e. it can be written $\mathbf{U} = \mathbf{Y}^\top \alpha$ where α are the dual variables. Substituting this into the primal formulation of PCA to obtain the dual, we get:

$$\begin{aligned}
\tilde{\boldsymbol{\Sigma}}_y \mathbf{Y}^\top \alpha &= \mathbf{Y}^\top \alpha \boldsymbol{\Lambda} \\
\mathbf{Y} \tilde{\boldsymbol{\Sigma}}_y \mathbf{Y}^\top \alpha &= \mathbf{Y} \mathbf{Y}^\top \alpha \boldsymbol{\Lambda} \\
\frac{1}{N} \mathbf{Y} \mathbf{Y}^\top \mathbf{Y} \mathbf{Y}^\top \alpha &= \mathbf{Y} \mathbf{Y}^\top \alpha \boldsymbol{\Lambda} \\
\frac{1}{N} \mathbf{Y} \mathbf{Y}^\top \alpha &= \alpha \boldsymbol{\Lambda} \qquad (2.68)
\end{aligned}$$

From (2.68) it can be seen that the principal axes $\mathbf{U}$ can be found in terms of the eigenvectors α of the inner product matrix $\mathbf{Y}\mathbf{Y}^\top$. The projection $\mathbf{x}_n$ of a data point $\mathbf{y}_n$ onto $\mathbf{U}$ is given by $\mathbf{x}_n = \mathbf{y}_n^\top \mathbf{Y}^\top \alpha$ i.e. in terms of inner products between the data points. This derivation is fundamental for implementing kernel PCA (Smola *et al.*, 1999, 2001; Schölkopf *et al.*, 1998, 1999).

Suppose that each data point is mapped into a feature space by a set of M functions $\phi : \mathbf{y}_n \rightarrow \phi(\mathbf{y}_n)$, and the inner products between the vectors in feature space are defined by the kernel $k(\mathbf{y}_i, \mathbf{y}_j) = \phi(\mathbf{y}_i)^\top \phi(\mathbf{y}_j)$. To perform PCA on the feature vectors, we require the eigenvectors $\mathbf{U}_\phi$ of the covariance matrix in feature space:

$$\Phi^\top \Phi \mathbf{U}_\phi = \mathbf{U}_\phi \Lambda \tag{2.69}$$

where we have defined $\Phi \in \Re^{N \times M}$ as the design matrix in feature space $\Phi = [\phi(\mathbf{y}_1)^\top, ..., \phi(\mathbf{y}_N)^\top]^\top$. Instead of using (2.69) which involves calculating each $\phi(\mathbf{y}_n)$ (which may be unknown) we can use the dual formulation of PCA in (2.68) and replace the inner product of the feature vectors $\Phi\Phi^\top$ with a kernel matrix (or Gram matrix) $\mathbf{K} \in \Re^{N \times N}$ where $\mathbf{K}_{ij} = k(\mathbf{y}_i, \mathbf{y}_j)$, and k is the kernel function, giving the kernel PCA eigenproblem:

$$\mathbf{K}\alpha \quad = \quad \alpha \boldsymbol{\Lambda} \tag{2.70}$$

Calculating the eigenvectors $\mathbf{U}_\phi = \Phi^\top \alpha$ of the covariance matrix in feature space involves calculating Φ, which may be unknown. Instead, we can calculate the projection $\mathbf{x}_*$ of a data point $\mathbf{y}_*$ onto $\mathbf{U}_\phi$ as $\mathbf{x}_* = k_*^\top \alpha$, where $k_* = [k(\mathbf{y}_*, \mathbf{y}_1), ..., k(\mathbf{y}_*, \mathbf{y}_N)]^\top$.

2.5 Modelling two data sets

In the previous sections we have reviewed graphical models, latent variable models, and techniques for finding a reduced dimensionality representation of a single data set, where our models were based on the graphical model shown in Figure 2.9. In this section, we show how these methods can be extended to modelling two data sets, and we also highlight some of the difficulties associated with these methods.

It is assumed there is some dependency between the two data sets $\mathbf{y}_1$ and $\mathbf{y}_2$ that we are trying to model. A key feature of methods that try to find interesting structure between two data sources is that some kind of dimensionality reduction is used; the modelling of the dependency is constrained by assuming that there is a reduced dimensionality representation of the relationship, which exists in some feature space $\mathbf{x}$. This allows the signal and noise subspaces to be separated. We denote the mapping of $\mathbf{y}_1$ and $\mathbf{y}_2$ to the feature space as $\mathbf{x}_1$ and $\mathbf{x}_2$ respectively. These sets of extracted features should reflect the information common to both data sets. In general, discriminative modelling approaches estimate the parameters of the mappings to the two sets of features to try to explicitly optimise some dependency criterion between $\mathbf{x}_1$ and $\mathbf{x}_2$, while generative modelling approaches are based on the estimation of the joint probability density $p(\mathbf{y}_1, \mathbf{y}_2)$ of the observed data, tuning the parameters of a model that would generate the observations. A key feature of the existing generative models for two data sources is the assumption of a shared latent variable $\mathbf{x}$ that underlies the data sources, and that the data sources are conditionally independent of each other, given the latent variable. After training the model, the latent space representations of each set are given by the posterior distributions $p(\mathbf{x} \mid \mathbf{y}_1)$ and $p(\mathbf{x} \mid \mathbf{y}_2)$. With both the discriminative and generative modelling approaches, the same problems exist:

- Defining the mappings from each data space to the shared feature space.
- Defining some dependency measure between the two sets of extracted features for optimisation.

Within the generative modelling framework, it is difficult to put constraints on the posterior distributions, and thus difficult to explicitly include some dependency measure. Instead we have to encode our prior knowledge about the two data sets as we see in Figure 2.9; i.e. structuring our model such that the two data sets interact only through a shared latent process. However, this does not guarantee that after observing the data, there will be strong dependency between the posterior distributions. Conversely, a discriminant model will explicitly try to optimise some dependency measure between the two extracted feature sets, but this can seem *ad hoc*, and since we do not define a full probability density over all the variables we cannot calculate quantities such as $p(\mathbf{y}_1 \mid \mathbf{y}_2)$ and $p(\mathbf{y}_2 \mid \mathbf{y}_1)$ for prediction.

2.5.1 An overview of discriminative techniques

A well established statistical technique for finding linearly correlated features between data sets is canonical correlation analysis (CCA) (Hotelling, 1936; Borga, 1998; Lai & Fyfe, 1999). Correlation is a good measure of dependency between signals because unlike covariance, it is invariant to the signal magnitudes. However, methods that rely on correlation have their limitations since they are based on second order statistics, which is only well justified for Gaussian distributed data. One way of extending CCA is by taking higher order statistics into account, which could be achieved by extending existing independent component analysis (ICA) algorithms to two data sets as in (Akaho *et al.*, 1999), (de Bie & de Moor, 2002). Kernel canonical correlation analysis was introduced in (Lai & Fyfe, 2000), where kernel functions implicitly define nonlinear transformation of the data sets into a feature space where linear CCA is performed.

Information theory offers a theoretical framework in which dependencies between variables can be analysed. Given two variables $\mathbf{x}_1$ and $\mathbf{x}_2$, a common measure of dependency is mutual information, which is a measure of the amount of information that $\mathbf{x}_1$ contains about $\mathbf{x}_2$ (and $\mathbf{x}_2$ contains about $\mathbf{x}_1$). It is defined as $I(\mathbf{x}_1;\mathbf{x}_2) = H(\mathbf{x}_1) - H(\mathbf{x}_1 \mid \mathbf{x}_2)$ (or also $H(\mathbf{x}_2) - H(\mathbf{x}_2 \mid \mathbf{x}_1)$) where $H(\mathbf{x}_1) = -\int p(\mathbf{x}_1)\log p(\mathbf{x}_1)d\mathbf{x}_1$ is the marginal entropy and $H(\mathbf{x}_1 \mid \mathbf{x}_2) = -\int\int p(\mathbf{x}_1 \mid \mathbf{x}_2)p(\mathbf{x}_2)\log p(\mathbf{x}_1 \mid \mathbf{x}_2)d\mathbf{x}_1 d\mathbf{x}_2$ is the conditional entropy. Given two signals that are expected to have a dependency on each other, from an information theoretic point of view this means that there should be features in the signals that have a high mutual information between them.

There are many methods in the literature that use ideas from information theory for the unsupervised modelling of a single data source. These methods use some measure based on the mutual information between the data $\mathbf{y}$ and its coded representation $\mathbf{x}$ such that $\mathbf{x}$ is informative about $\mathbf{y}$. This was first introduced to the machine learning field as the principle of maximum information preservation (Infomax) in (Linsker, 1988). However, computing mutual information exactly can be difficult since it requires probability densities over the variables in question and involves integration over functions of the densities. As a result, a lot of the methods that use mutual information specify the forms of the densities such that the required calculations are analytically tractable. Generally, jointly Gaussian distributions are chosen (such as in (Linsker, 1988)), but unfortunately this can result in loss of modelling power due to the oversimplification of the statistical relationship between variables, and restriction to linear mappings between the data and codes. A number of methods have been proposed to extend Infomax to arbitrary densities and (possibly) nonlinear mappings by using Parzen window density estimation to directly estimate the required entropies, such as in (Viola, 1995), and the Information Theoretic Learning framework of Principe et al (Principe *et al.*, 2000). A different approach is taken in (Agakov, 2005; Agakov & Barber, 2004), in which a family of variational lower bounds on mutual information between the data and its coded representation is introduced to give a theoretically rigorous approach to information preservation.

One problem with using mutual information for an unsupervised learning problem is that it does not explicitly define which parts of the information are useful. One way to extract ‘useful’ information is by specifying some prefixed architecture for the model to implicitly define some measure of usefulness. Another way of constraining the extracted information is by approaching the problem from a semisupervised perspective, in which another variable, which signifies what parts of the information in the data is relevant, is used to guide the feature extraction. Examples of using mutual information in a semisupervised setting are the feature extraction algorithms of (Torkkola, 2003), in which the mutual information between class labels and the transformed data is maximised, and the family of Information Bottleneck (IB) methods (Tishby *et al.*, 1999) which maximise the amount of information that the compressed representation $\mathbf{x}$ of a data variable $\mathbf{y}$ contains about some relevant variables $\mathbf{t}$, while minimising the information between the compressed representation and the data. This can be stated formally as the minimisation of the Lagrangian $I(\mathbf{x};\mathbf{y}) - \beta I(\mathbf{x};\mathbf{t})$.

The problem of extracting features from two related data sources is similar to the semi-supervised information preservation problem for a single data source. Whereas the latter methods use an additional variable to indicate which features in the data is useful, learning a representation for two data sources $\mathbf{y}_1$ and $\mathbf{y}_2$ uses $\mathbf{y}_1$ to guide the feature extraction for $\mathbf{y}_2$, and vice versa, such that each data variable acts as the relevance variable for the other. For this reason, semi-supervised information theoretic frameworks for feature extraction could be extended to our problem of modelling two data sources. Two interesting extensions to the IB framework are relevant to our problem. Whereas the original framework was based on a single sided principle, in that only the data variable and not the relevance variable is compressed, in (Friedman *et al.*, 2001) a symmetric form of the problem is proposed such that both variables are compressed. Given two data variables $\mathbf{y}_1$ and $\mathbf{y}_2$, $\mathbf{y}_1$ is compressed into $\mathbf{x}_1$ and $\mathbf{y}_2$ into $\mathbf{x}_2$ such that $\mathbf{x}_1$ extracts the information $\mathbf{y}_1$ contains about $\mathbf{y}_2$, and at the same time $\mathbf{x}_2$ extracts the information $\mathbf{y}_2$ contains about $\mathbf{y}_1$. This is achieved through minimising the Lagrangian: $I(\mathbf{x}_1;\mathbf{y}_1)+I(\mathbf{x}_2;\mathbf{y}_2)-\gamma I(\mathbf{x}_1;\mathbf{x}_2)$. Another extension of the IB framework is the extension to continuous variables in (Chechik *et al.*, 2003), (Chechik & Globerson, 2003), in contrast with earlier work which focused on categorical variables. By assuming that the data variable $\mathbf{y}$ and the relevance variable $\mathbf{t}$ are jointly multivariate Gaussian variables, $\mathbf{y}$ is compressed via a linear transformation into $\mathbf{x}$ while preserving information about $\mathbf{t}$. The analytic closed form solution of the optimal linear projection is shown to be the canonical basis vectors (from CCA) for $\mathbf{y}$ and $\mathbf{t}$.

Several methods have been proposed specifically for learning from two data sources using mutual information. In this context, the mutual information is maximised between the coded representations of the data sources. Becker and Hinton presented Imax in (Becker, 1992; Becker & Hinton, 1992; Becker, 1996), a variant of Infomax, which aims to maximise the information between outputs of two neighbouring neural networks. This architecture can be used to extract spatially coherent features in simulations of visual processing. A similar approach is presented in (Kay, 1992).

As we detailed above, methods for the analysis of two data sources using mutual information suffer from complications, due to the difficulties in calculating mutual information. Another complication exists in the constraining of the model. Suppose that we have two data sources $\mathbf{y}_1$ and $\mathbf{y}_2$, from which we want to extract features (or a new representation) $\mathbf{x}_1$ and $\mathbf{x}_2$ respectively, that have maximum mutual information. The extracted feature sets would be expected to be a compact representation of the relationship between the two data sources. However this may not be the case; we also desire the joint entropy $H(\mathbf{x}_1, \mathbf{x}_2)$ to be small, i.e. we want to minimise the conditional entropies $H(\mathbf{x}_1|\mathbf{x}_2)$ and $H(\mathbf{x}_2|\mathbf{x}_1)$ such that the features only capture the common information between $\mathbf{y}_1$ and $\mathbf{y}_2$.

This problem of finding 'efficient' features was addressed in (Butz & Thiran, 2005) by introducing the feature efficiency coefficient which both maximises the mutual information between features and minimises the joint entropy, given by:

$$e(\mathbf{x}_1, \mathbf{x}_2) = \frac{I(\mathbf{x}_1, \mathbf{x}_2)}{H(\mathbf{x}_1, \mathbf{x}_2)} \tag{2.71}$$

Since $H(\mathbf{x}_1, \mathbf{x}_2) \geq I(\mathbf{x}_1, \mathbf{x}_2)$ and both terms are positive, $0 \leq e(\mathbf{x}_1, \mathbf{x}_2) \leq 1$. For highly efficient features, $e(\mathbf{x}_1, \mathbf{x}_2)$ should be close to 1. A similar functional called normalised entropy (Studholme *et al.*, 1999) is used in the field of multi-modal medical image registration.

2.5.2 An overview of generative techniques

While there are many discriminative techniques for modelling two data sources, there are comparatively few generative techniques. Some possible generative models of two data sets are represented by the graphical models in Figure 2.9. Figure 2.9a shows the two observed data variables $\mathbf{y}_1$ and $\mathbf{y}_2$ and their relationship; modelling the two data sources is equivalent to estimating their joint distribution $p(\mathbf{y}_1, \mathbf{y}_2)$. Direct estimation of this joint distribution is problematic, particularly if $\mathbf{y}_1$ and $\mathbf{y}_2$ are high dimensional, thus it is necessary to further constrain the model. In Figures 2.9b and 2.9c we enforce a prior structure on our data which assumes that the data sets share a common underlying source $\mathbf{x}$, and also that the data sets are conditionally independent of each other:

$$p(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}) = p(\mathbf{y}_1 \mid \mathbf{x}) p(\mathbf{y}_2 \mid \mathbf{x}) \tag{2.72}$$

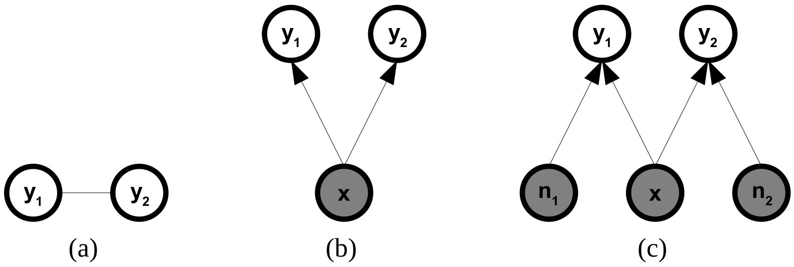


Figure 2.9: Possible graphical models for modelling two data sources.

Figure 2.9b represents our intuition that the dependency between $\mathbf{y}_1$ and $\mathbf{y}_2$ is due to their being different manifestations of the same underlying process. This is represented by the hidden (shown by the grey shade), or **latent variable** $\mathbf{x}$. An alternative graphical model is shown in (c). This model explicitly represents the 'private' information associated with each sensor by a random variable. Both (b) and (c) constrain the joint distribution such that it has fewer degrees of freedom before it is directly estimated from the data. After training the model, we can apply Bayes rule to calculate quantities such as:

$$p(\mathbf{x} \mid \mathbf{y}_1) = \frac{p(\mathbf{y}_1 \mid \mathbf{x})p(\mathbf{x})}{p(\mathbf{y}_1)}, p(\mathbf{x} \mid \mathbf{y}_2) = \frac{p(\mathbf{y}_2 \mid \mathbf{x})p(\mathbf{x})}{p(\mathbf{y}_2)} \tag{2.73}$$

the low dimensional representations of each data source, and the predictive distributions over one data set given the other:

$$p(\mathbf{y}_1 \mid \mathbf{y}_2) = \frac{p(\mathbf{y}_1, \mathbf{y}_2)}{p(\mathbf{y}_2)}, \qquad p(\mathbf{y}_2 \mid \mathbf{y}_1) = \frac{p(\mathbf{y}_1, \mathbf{y}_2)}{p(\mathbf{y}_1)} \tag{2.74}$$

These models serve as a good basis for modelling dependencies with generative models; some models that already exist in the literature can be placed within this framework. One recent technique is the probabilistic canonical correlation analysis model (PCCA) in (Bach & Jordan, 2005), which places CCA in a Gaussian density estimation framework with the model structure as in Figure 2.9b.

There are various extensions to this model. In (Archambeau *et al.*, 2006), the Gaussian densities are replaced with Student-t densities to create a model that is more robust to outliers, and a variational Bayesian version is proposed in (Wang, 2007) (in the same spirit as the variational Bayesian extension (Bishop, 1999) of Probabilistic PCA) which allows the dimensionality of the latent space (and hence the effective number of canonical correlations) to be determined automatically. These methods assume a linear relationship between each data variable and its corresponding set of features, and consequently models the relationship between the two data sets as linear.

One feature of the PCCA model is that each data source is modelled as the sum of two independent components: $\mathbf{y}_1 = \mathbf{f}_1 + \mathbf{n}_1, \mathbf{y}_2 = \mathbf{f}_2 + \mathbf{n}_2$, where $\mathbf{n}_1$ and $\mathbf{n}_2$ are noise components which model the within-set variation, $\mathbf{f}_1$ and $\mathbf{f}_2$ are components which are linearly related to a shared latent variable $\mathbf{x}$ and model the between-sets variation, and $\mathbf{x}, \mathbf{n}_1$ and $\mathbf{n_2}$ are independent of each other. The structure shown in Figure 2.9c is implicit in the PCCA model, and as noted in (Klami & Kaski, 2006), it is necessary for each noise component to be flexible enough to completely model the marginal density of its corresponding data variable, and hence all of the within-set variation. This allows the other 'shared' components to solely model the between-set variation, since none of their modelling capacity is wasted on modelling the variation within the sets. This can be thought of constraining the model in such as way to find efficient features (as we saw in (2.71)) that only represent shared information between the data sources.

It is difficult to extend this idea to more complicated models; specifying the noise components to completely model all the within-set variation is difficult when the data follows a more complex distribution than a unimodal exponential family distribution. Another complication is that when considering nonlinear relationships between the data space and the latent space, the noise and shared components may not be independent. However, there are a few nonlinear extensions of canonical correlation analysis that are formulated as generative models.

In (Verbeek *et al.*, 2004), the authors propose a nonlinear canonical correlation analysis method. The two data sets are assumed to come from separate nonlinear manifolds that share an underlying global coordinate system, where each manifold is modelled by a mixture of aligned local models. Interestingly, the method is different from standard mixture models in that it integrates local feature extractors into a single global representation in the spirit of (Roweis *et al.*, 2002). The global coordination of the local models is achieved by adding a regularizer term to the standard maximum likelihood objective function, similar to a variational approach. However, this model does not model the within-set variation, and instead assumes that the data lies close to each nonlinear manifold.

Another approach to nonlinear canonical correlation analysis would be to use a different specification of the nonlinear relationship between the data and the latent space. Instead of modelling the nonlinear relationship by a mixture of aligned local models, an alternative is to specify a global nonlinear mapping, for instance by placing a Gaussian process prior over the space of nonlinear functions of the latent variables. The Gaussian process regression framework is extended in (Boyle & Frean, 2005a,b) to handle multiple coupled outputs by assuming that dependent outputs are related through a shared latent process, and the variation within an output is modelled by a separate latent process, following the structure in Figure 2.9c. However, this model is formulated for regression problems and assumes that the latent coordinates are known. In the next section, we review canonical correlation analysis and its different variants, and use it as a starting point for creating dependency seeking generative models.

2.5.3 Canonical correlation analysis

Canonical correlation analysis (CCA) (Hotelling, 1936) proposes a way for dimensionality reduction by taking the relationship between two sets of variables into account. CCA is concerned with finding linear relationships between the two sets of variables. Given two sets of zero mean data variables $\mathbf{y}_1 \in \Re^{m_1}$ and $\mathbf{y}_2 \in \Re^{m_2}$, where m_1 and m_2 are the dimensions of $\mathbf{y}_1$ and $\mathbf{y}_2$ respectively, CCA finds linear projections of each variable $\mathbf{x}_1 = \mathbf{U}_1^\top \mathbf{y}_1$ and $\mathbf{x}_2 = \mathbf{U}_2^\top \mathbf{y}_2$, termed the canonical variates, such that the correlation between $\mathbf{x}_1$ and $\mathbf{x}_2$ is maximised, and $\mathbf{U}_1 \in \Re^{m_1 \times q}$ and $\mathbf{U}_2 \in \Re^{m_2 \times q}$, where $q \leq \min(m_1, m_2)$, are matrices whose columns $\mathbf{U}_{1,i}, \mathbf{U}_{2,i}, i = 1, .., q$ form the q pairs of canonical vectors. We can find $\mathbf{U}_1$ and $\mathbf{U}_2$ as the eigenvectors of the generalised eigenvalue problem:

$$\begin{pmatrix} 0 & \tilde{\mathbf{\Sigma}}_{12} \\ \tilde{\mathbf{\Sigma}}_{21} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{\Sigma}}_{11} & 0 \\ 0 & \tilde{\mathbf{\Sigma}}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{pmatrix} \rho \tag{2.75}$$

where ρ is the diagonal matrix of canonical correlations, and

$$\tilde{\mathbf{\Sigma}} = E\left(\begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix}^\top \right) = \begin{pmatrix} \tilde{\mathbf{\Sigma}}_{11} & \tilde{\mathbf{\Sigma}}_{12} \\ \tilde{\mathbf{\Sigma}}_{21} & \tilde{\mathbf{\Sigma}}_{22} \end{pmatrix} \tag{2.76}$$

This can also be formulated as a symmetric eigenvalue problem:

$$\begin{pmatrix} 0 & \tilde{\mathbf{\Sigma}}_{11}^{-\frac{1}{2}} \tilde{\mathbf{\Sigma}}_{12} \tilde{\mathbf{\Sigma}}_{22}^{-\frac{1}{2}} \\ \tilde{\mathbf{\Sigma}}_{22}^{-\frac{1}{2}} \tilde{\mathbf{\Sigma}}_{21} \tilde{\mathbf{\Sigma}}_{11}^{-\frac{1}{2}} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{pmatrix} \rho \tag{2.77}$$

where $\mathbf{V}_1 = \tilde{\mathbf{\Sigma}}_{11}^{\frac{1}{2}} \mathbf{U}_1$ and $\mathbf{V}_2 = \tilde{\mathbf{\Sigma}}_{22}^{\frac{1}{2}} \mathbf{U}_2$. Another property of CCA is that the projections onto canonical directions corresponding to a different canonical correlation are uncorrelated such that $\mathbf{U}_1^\top \tilde{\mathbf{\Sigma}}_{11} \mathbf{U}_1 = \mathbf{I}_{m_1}$ and $\mathbf{U}_2^\top \tilde{\mathbf{\Sigma}}_{22} \mathbf{U}_2 = \mathbf{I}_{m_2}$. Canonical correlation analysis is also related to mutual information. If $\mathbf{y}_1$ and $\mathbf{y}_2$ are jointly Gaussian distributed, then the mutual information between $\mathbf{y}_1$ and $\mathbf{y}_2$ is given by the sum of the mutual information between the canonical variates $\mathbf{x}_1$ and $\mathbf{x}_2$:

$$I(\mathbf{y}_1; \mathbf{y}_2) = \frac{1}{2} \log \left(\frac{1}{\prod_i (1 - \rho_i^2)} \right) = \frac{1}{2} \sum_i \log \left(\frac{1}{(1 - \rho_i^2)} \right) \tag{2.78}$$

2.5.4 Probabilistic Canonical Correlation Analysis

Canonical correlation analysis (CCA) was formulated as a Gaussian latent variable model in (Bach & Jordan, 2005). It is found that the posterior distributions of the latent variables lie in the same linear subspaces as those defined by standard CCA. Using the definition for the Gaussian latent variable model from Section 2.2.1, $\mathbf{y}$ is defined as the concatenation of two sets of data variables i.e. $\mathbf{y} = [\mathbf{y}_1^\top, \mathbf{y}_2^\top]^\top$, where $\mathbf{y}_1 \in \Re^{m_1}, \mathbf{y}_2 \in \Re^{m_2}$ with m_1 and m_2 being the dimensions of the two data variable sets and $\boldsymbol{\mu} = [\boldsymbol{\mu}_1^\top, \boldsymbol{\mu}_2^\top]^\top$, where $\boldsymbol{\mu}_1 \in \Re^{m_1}, \boldsymbol{\mu}_2 \in \Re^{m_2}$. $\mathbf{W} = [\mathbf{W}_1^\top, \mathbf{W}_2^\top]^\top$ with $\mathbf{W}_1 \in \Re^{m_1 \times q}, \mathbf{W}_2 \in \Re^{m_2 \times q}$, and $\mathbf{x}_n \in \Re^q$ is the shared latent variable for the nth pair of data variables $\mathbf{y}_n$. The noise covariance matrix is constrained to be of block diagonal form:

$$\boldsymbol{\Sigma}_{\mathbf{n}} = \begin{pmatrix} \boldsymbol{\Psi}_1 & 0 \\ 0 & \boldsymbol{\Psi}_2 \end{pmatrix} \tag{2.79}$$

where $\boldsymbol{\Psi}_1 \in \Re^{m_1 \times m_1}, \boldsymbol{\Psi}_2 \in \Re^{m_2 \times m_2}$ The maximum likelihood solutions for the parameters are given by:

$$\hat{\boldsymbol{\mu}}_1 = \tilde{\boldsymbol{\mu}}_1 \tag{2.80}$$

$$\hat{\boldsymbol{\mu}}_2 = \tilde{\boldsymbol{\mu}}_2 \tag{2.81}$$

$$\hat{\mathbf{W}}_1 = \tilde{\boldsymbol{\Sigma}}_{11} \mathbf{U}_{1q} \mathbf{P}_q \mathbf{R} \tag{2.82}$$

$$\hat{\mathbf{W}}_2 = \tilde{\boldsymbol{\Sigma}}_{22} \mathbf{U}_{2q} \mathbf{P}_q \mathbf{R} \tag{2.83}$$

$$\hat{\boldsymbol{\Psi}}_1 = \tilde{\boldsymbol{\Sigma}}_{11} - \hat{\mathbf{W}}_1 \hat{\mathbf{W}}_1^\top \tag{2.84}$$

$$\hat{\boldsymbol{\Psi}}_2 = \tilde{\boldsymbol{\Sigma}}_{22} - \hat{\mathbf{W}}_2 \hat{\mathbf{W}}_2^\top \tag{2.85}$$

where $\tilde{\boldsymbol{\mu}}_1$ and $\tilde{\boldsymbol{\mu}}_2$ are the sample means of the two sets of data variables. $\mathbf{U}_{1q} \in \Re^{m_1 \times q}$ and $\mathbf{U}_{2q} \in \Re^{m_2 \times q}$ are matrices whose columns consist of the first q canonical directions for $\mathbf{y}_1$ and $\mathbf{y}_2$ respectively, $\mathbf{P}_q$ is the diagonal matrix of the q largest canonical correlations, $\mathbf{R} \in \Re^{q \times q}$ is a rotation matrix, and we have defined $E(\mathbf{y}\mathbf{y}^\top) = E(\binom{\mathbf{y}_1}{\mathbf{y}_2}\binom{\mathbf{y}_1}{\mathbf{y}_2}^\top) = \begin{pmatrix} \tilde{\boldsymbol{\Sigma}}_{11} & \tilde{\boldsymbol{\Sigma}}_{12} \\ \tilde{\boldsymbol{\Sigma}}_{21} & \tilde{\boldsymbol{\Sigma}}_{22} \end{pmatrix}$

2.5.5 Kernel CCA

A kernel variant of canonical correlation analysis has been proposed in (Bach & Jordan, 2002; Lai & Fyfe, 2000), where kernel functions implicitly define a nonlinear transformation of the two data sources into a feature space where linear CCA is performed. This allows us to find nonlinear relationships between the two sets of data variables.

Canonical correlation analysis is conventionally defined in terms of the covariance matrices of the two data variables $\mathbf{y}_1$ and $\mathbf{y}_2$, as we reviewed in Section 2.5.3: we can find $\mathbf{W}_1$ and $\mathbf{W}_2$, the canonical vectors, as the eigenvectors of the generalised eigenvalue problem:

$$\begin{pmatrix} 0 & \tilde{\mathbf{\Sigma}}_{12} \\ \tilde{\mathbf{\Sigma}}_{21} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{\Sigma}}_{11} & 0 \\ 0 & \tilde{\mathbf{\Sigma}}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \end{pmatrix} \rho \tag{2.86}$$

where ρ is the diagonal matrix of canonical correlations, and

$$\tilde{\mathbf{\Sigma}} = E\left(\begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix}^\top \right) = \begin{pmatrix} \tilde{\mathbf{\Sigma}}_{11} & \tilde{\mathbf{\Sigma}}_{12} \\ \tilde{\mathbf{\Sigma}}_{21} & \tilde{\mathbf{\Sigma}}_{22} \end{pmatrix} = \frac{1}{N} \begin{pmatrix} \mathbf{Y}_1^\top \mathbf{Y}_1 & \mathbf{Y}_1^\top \mathbf{Y}_2 \\ \mathbf{Y}_2^\top \mathbf{Y}_1 & \mathbf{Y}_2^\top \mathbf{Y}_2 \end{pmatrix} \tag{2.87}$$

where $\mathbf{Y}_1 = [\mathbf{y}_{1,1}, ..., \mathbf{y}_{1,N}]^\top$ and $\mathbf{Y}_2 = [\mathbf{y}_{2,1}, ..., \mathbf{y}_{2,N}]^\top$. To obtain the dual of (2.86), it is noted that the canonical vectors $\mathbf{W}_1$ and $\mathbf{W}_2$ can be written as:

$$\mathbf{W}_1 = \mathbf{Y}_1^\top \alpha_1 \tag{2.88}$$

$$\mathbf{W}_2 = \mathbf{Y}_2^\top \alpha_2 \tag{2.89}$$

Substituting into the primal equations for CCA given in (2.86), we get:

$$\begin{aligned} \begin{pmatrix} 0 & \tilde{\mathbf{\Sigma}}_{12}\mathbf{Y}_2^\top \\ \tilde{\mathbf{\Sigma}}_{21}\mathbf{Y}_1^\top & 0 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} &= \begin{pmatrix} \tilde{\mathbf{\Sigma}}_{11}\mathbf{Y}_1^\top & 0 \\ 0 & \tilde{\mathbf{\Sigma}}_{22}\mathbf{Y}_2^\top \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} \rho \\ \begin{pmatrix} 0 & \mathbf{Y}_1\tilde{\mathbf{\Sigma}}_{12}\mathbf{Y}_2^\top \\ \mathbf{Y}_2\tilde{\mathbf{\Sigma}}_{21}\mathbf{Y}_1^\top & 0 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} &= \begin{pmatrix} \mathbf{Y}_1\tilde{\mathbf{\Sigma}}_{11}\mathbf{Y}_1^\top & 0 \\ 0 & \mathbf{Y}_2\tilde{\mathbf{\Sigma}}_{22}\mathbf{Y}_2^\top \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} \rho \end{aligned} \tag{2.90}$$

The dual problem for CCA is given in (2.90) which is given in terms of the inner products $\mathbf{Y}_1\mathbf{Y}_1^\top$ and $\mathbf{Y}_2\mathbf{Y}_2^\top$ (which can be easily seen through the substitutions for the different blocks for $\tilde{\mathbf{\Sigma}}$). The canonical vectors $\mathbf{W}_1$ and $\mathbf{W}_2$ can be recovered from the dual variables by applying (2.88) and (2.89). The canonical variates $\mathbf{x}_{1,*}$ and $\mathbf{x}_{2,*}$

for a pair of test points $\mathbf{y}_{1,*}$ and $\mathbf{y}_{2,*}$ can also be found in terms of the dual variables: $\mathbf{x}_{1,*} = \mathbf{y}_{1,*}^\top \mathbf{Y}_1^\top \alpha_1$ and $\mathbf{x}_{2,*} = \mathbf{y}_{2,*}^\top \mathbf{Y}_2^\top \alpha_2$.

Suppose that both sets of data variables $\mathbf{y}_{1,n}$ and $\mathbf{y}_{2,n}$ are mapped to (possibly different) feature spaces by a set of functions $\phi_1 : \mathbf{y}_{1,n} \rightarrow \phi_1(\mathbf{y}_{1,n})$ and $\phi_2 : \mathbf{y}_{2,n} \rightarrow \phi_2(\mathbf{y}_{2,n})$, where the inner products between the vectors in feature space are defined by the kernel functions $k_1(\mathbf{y}_{1,i}, \mathbf{y}_{1,j}) = \phi_1(\mathbf{y}_{1,i})^\top \phi_1(\mathbf{y}_{1,j})$ and $k_2(\mathbf{y}_{2,i}, \mathbf{y}_{2,j}) = \phi_2(\mathbf{y}_{2,i})^\top \phi_2(\mathbf{y}_{2,j})$. Defining $\Phi_1 = [\phi_1(\mathbf{y}_{1,1}), ..., \phi_1(\mathbf{y}_{1,N})]^\top$ and $\Phi_2 = [\phi_2(\mathbf{y}_{2,1}), ..., \phi_2(\mathbf{y}_{2,N})]^\top$ as $\mathbf{Y}_1$ and $\mathbf{Y}_2$ mapped into their respective feature spaces, and exploiting the dual formulation of CCA given in (2.90), kernel Canonical Correlation Analysis can be formulated as:

$$\begin{pmatrix} 0 & \mathbf{K}_1\mathbf{K}_2 \\ \mathbf{K}_2\mathbf{K}_1 & 0 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} \mathbf{K}_1^2 & 0 \\ 0 & \mathbf{K}_2^2 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} \rho \tag{2.91}$$

where $\mathbf{K}_1 = \Phi_1\Phi_1^\top \in \Re^{N \times N}$ and $\mathbf{K}_2 = \Phi_2\Phi_2^\top \in \Re^{N \times N}$ are the kernel matrices where $\mathbf{K}_{1(i,j)} = k_1(\mathbf{y}_{1,i}, \mathbf{y}_{1,j})$ and $\mathbf{K}_{2(i,j)} = k_2(\mathbf{y}_{2,i}, \mathbf{y}_{2,j})$. To calculate the canonical variates $\mathbf{x}_{1,*}$ and $\mathbf{x}_{2,*}$ i.e. the projections of a pair of test points $\mathbf{y}_{1,*}$ and $\mathbf{y}_{2,*}$ onto their respective canonical vectors $\mathbf{W}_{1,\phi}$ and $\mathbf{W}_{2,\phi}$ (which are generally not known), we use

$$\mathbf{x}_{1,*} = k_{1,*}^\top \alpha_1, \quad \mathbf{x}_{2,*} = k_{2,*}^\top \alpha_2 \tag{2.92}$$

where $k_{1,*} = [k_1(\mathbf{y}_{1,*}, \mathbf{y}_{1,1}), ..., k_1(\mathbf{y}_{1,*}, \mathbf{y}_{1,N})]^\top$ and
$k_{2,*} = [k_2(\mathbf{y}_{2,*}, \mathbf{y}_{2,1}), ..., k_2(\mathbf{y}_{2,*}, \mathbf{y}_{2,N})]^\top$.

2.6 Summary

In this chapter, we have outlined the problem of learning from two data sources, and reviewed the probabilistic approach that we will be using for finding common features. We discussed the relative merits of using both generative and discriminative probabilistic models. Though discriminative techniques may be more efficient for finding a joint representation for two data sources since this involves directly optimising a measure of similarity between the extracted features, a probability density is not defined over the variables of the problem. We consider generative models to be more appropriate for describing the joint structure between two related data sources, since this represents the data generation process, and we can resynthesise different configurations from it such as the predictive distribution over one source given the other.

We also reviewed a number of parametric and nonparametric Bayesian methods for finding the underlying structure of both one and two data sources. Existing linear models for modelling two sources are canonical correlation analysis (CCA), and probabilistic CCA, and an existing nonlinear model is Kernel CCA. Since the problem of finding nonlinearly related features between two data sets is ill posed, because it is possible to find spurious correlations, in the rest of the thesis we propose that using a probabilistic generative approach is the preferred solution. We also use nonparametric Bayesian methods due to their flexibility and their ability to automatically determine model complexity from the data. The work in this chapter provides the background to the rest of the thesis which contains our own research in this area.

Chapter 3

Generative models for finding shared structure

3.1 Introduction

In this chapter, we describe some generative models for finding dependencies between two data sets. In general, most methods that seek dependencies between two data sets are discriminative methods, which aim to extract a set of features for each data set such that some dependency measure between the features is maximised. Although this can be effective since the modelling power is explicitly focused on finding dependent features between the two data sets, discriminative methods can seem *ad hoc*. In particular, for two data sets that have a complex (possibly nonlinear) relationship, it is problematic to choose how the data is mapped into the shared feature space. Another drawback of the discriminative approach is that a probability density is not defined and it is not clear how to predict densities of one data set given the other. Using generative models for seeking structure between data sets is appealing since a probability density is defined for the data sets, allowing us to calculate predictive densities and to determine the parameters (or hyperparameters) of the mappings in a principled way within a probabilistic framework. It is also possible to insert prior knowledge about the underlying shared process into the model.

Probabilistic generative models of two related sets of data variables describe the shared features as a shared latent variable underlying both sets. By defining the two data variables as conditionally independent given the latent variable, the latent variable is the only shared component of the data, and therefore should represent the common information. An example of a generative model for finding shared structure is probabilistic canonical correlation analysis (PCCA) (Bach & Jordan, 2005), which we reviewed in Section 2.5.4. PCCA models each data set as being linearly related to the underlying shared latent space i.e. each data dimension is a linear function of the latent variable. Because PCCA only defines linear projections of the data sets, the scope of its application is limited since it cannot accurately model data sets that have nonlinearly related shared features. An approach to create a nonlinear version of PCCA is to consider nonlinear functions of the latent variable, in the spirit of the generative topographic mapping (Bishop *et al.*, 1996), to create global nonlinear mappings between the latent and data spaces. However, the problem with this approach lies in specifying the function so that it is appropriate for the data, a common problem for parametric modelling approaches. We turn to nonparametric Bayesian methods which offer a way to define flexible priors over data sets; we use Gaussian processes (O'Hagan, 1978; Rasmussen & Williams, 2006) as prior distributions over the functions from latent to data space, inspired by the Gaussian process latent variable model (GPLVM) (Lawrence, 2004, 2005).

The work described here follows from (Leen & Fyfe, 2006) which describes a derivation of a dependency seeking generative model using linear mixtures of underlying Gaussian processes. The model defines a probabilistic relationship between two sets of data variables by assuming that the shared structure can be represented by a shared underlying latent variable, which acts as input to the Gaussian process priors over the shared (nonlinear) functions underlying the data. The resulting model is a probabilistic interpretation of nonlinear canonical correlation analysis, which we call GPLVM-CCA. In Section 3.2 we analyse the dependencies between two correlated data variables from an information theoretic perspective, and use these results to determine the structure of a dependency seeking generative model. We model each data source as a sum of two independent components, one which models the *shared information* between the two data sets, and one which models the *private information* contained within each source. In Section 3.3 we study linear generative models for finding dependencies, and derive an alternative interpretation of probabilistic canonical correlation analysis (PCCA). In Section 3.5 we use this alternative interpretation of PCCA to derive a probabilistic model of nonlinear PCCA, by integrating over the linear mappings between latent and data space to create Gaussian process 'mappings' over the data space. This places nonparametric priors over the underlying functions of the two data sets. In Section 3.6 we apply the GPLVM-CCA to a range of data sets, including a large scale image data set, and present the results. We demonstrate the way in which the GPLVM-CCA model can be used to learn a shared latent structure for both data sets, and for finding a predictive distribution over one data set given the other, even in the presence of missing values.

3.2 Analysing the dependencies between two data variables

In this section we study the dependencies between two correlated data variables from an information theoretic perspective (Shannon, 1948) to give us some insight into the construction of generative models for dependency analysis. Given two correlated data variables $\mathbf{y}_1$ and $\mathbf{y}_2$, we can visualise the way in which their joint entropy $H(\mathbf{y}_1, \mathbf{y}_2)$ can be broken down in Figure 3.1, following similar diagrams in (MacKay, 2003). The quantities of interest are the joint entropy $H(\mathbf{y}_1, \mathbf{y}_2)$, the marginal entropies $H(\mathbf{y}_1)$, $H(\mathbf{y}_2)$, the conditional entropies $H(\mathbf{y}_1 \mid \mathbf{y}_2)$, $H(\mathbf{y}_2 \mid \mathbf{y}_1)$, and the mutual information $I(\mathbf{y}_1; \mathbf{y}_2)$, which are defined as follows. The joint entropy of $\mathbf{y}_1$ and $\mathbf{y}_2$ is given by:

$$H(\mathbf{y}_1, \mathbf{y}_2) = -\int p(\mathbf{y}_1, \mathbf{y}_2) \log p(\mathbf{y}_1, \mathbf{y}_2) d\mathbf{y}_1 d\mathbf{y}_2 \tag{3.1}$$

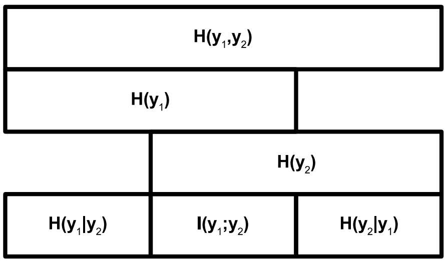


Figure 3.1: The relationship between joint entropy $H(\mathbf{y}_1, \mathbf{y}_2)$, marginal entropy $H(\mathbf{y}_1)$ and $H(\mathbf{y}_2)$, conditional entropy $H(\mathbf{y}_1 \mid \mathbf{y}_2)$ and $H(\mathbf{y}_2 \mid \mathbf{y}_1)$, and mutual information $I(\mathbf{y}_1; \mathbf{y}_2)$ for two correlated variables $\mathbf{y}_1$ and $\mathbf{y}_2$, where the relationships between the quantities is indicated by the relative area of the blocks.

The conditional entropy of $\mathbf{y}_1$ given $\mathbf{y}_2$, and the conditional entropy of $\mathbf{y}_2$ given $\mathbf{y}_1$ are given by:

$$H(\mathbf{y}_1 \mid \mathbf{y}_2) = -\int p(\mathbf{y}_1, \mathbf{y}_2) \log p(\mathbf{y}_1 \mid \mathbf{y}_2) d\mathbf{y}_1 d\mathbf{y}_2 \tag{3.2}$$

$$H(\mathbf{y}_2 \mid \mathbf{y}_1) = -\int p(\mathbf{y}_1, \mathbf{y}_2) \log p(\mathbf{y}_2 \mid \mathbf{y}_1) d\mathbf{y}_1 d\mathbf{y}_2 \tag{3.3}$$

The marginal entropies of $\mathbf{y}_1$ and $\mathbf{y}_2$ are given by:

$$H(\mathbf{y}_1) = -\int p(\mathbf{y}_1) \log p(\mathbf{y}_1) d\mathbf{y}_1 \tag{3.4}$$

$$H(\mathbf{y}_2) = -\int p(\mathbf{y}_2) \log p(\mathbf{y}_2) d\mathbf{y}_2 \tag{3.5}$$

and the mutual information between $\mathbf{y}_1$ and $\mathbf{y}_2$ is given by:

$$I(\mathbf{y}_1; \mathbf{y}_2) = H(\mathbf{y}_1) - H(\mathbf{y}_1 \mid \mathbf{y}_2) \tag{3.6}$$

$$= H(\mathbf{y}_2) - H(\mathbf{y}_2 \mid \mathbf{y}_1) \tag{3.7}$$

Some useful identities, which can be derived through manipulation of the previous equations, are as follows:

$$H(\mathbf{y}_1) = H(\mathbf{y}_1 \mid \mathbf{y}_2) + I(\mathbf{y}_1; \mathbf{y}_2) \tag{3.8}$$

$$H(\mathbf{y}_2) = H(\mathbf{y}_2 \mid \mathbf{y}_1) + I(\mathbf{y}_1; \mathbf{y}_2) \tag{3.9}$$

3.2.1 A generative process

From (3.8) and (3.9) of the previous section, it can be seen that the information content of each data variable (the marginal entropy) can be viewed as the sum of two independent components: a *shared information* with the other data variable (the mutual information between $\mathbf{y}_1$ and $\mathbf{y}_2$) and a *private information* (the conditional entropy). We also note that the two sets of private information are independent of each other since all the joint information is contained in the shared component.

In order to create a generative model of two correlated data variables $\mathbf{y}_1$ and $\mathbf{y}_2$, we suppose that they are generated according to:

$$\mathbf{y}_1 = \mathbf{f}_1 + \mathbf{n}_1 \tag{3.10}$$

$$\mathbf{y}_2 = \mathbf{f}_2 + \mathbf{n}_2 \tag{3.11}$$

such that each data variable consists of two independent components, $\mathbf{f}$, which models the shared information between the two data sources, and $\mathbf{n}$, an $\mathbf{f}$-independent noise process which models the private information. See Figure 3.2.

3.2.1.1 Modelling the shared information

To model the shared information, a latent variable $\mathbf{x}$ underlying $\mathbf{f}_1$ and $\mathbf{f}_2$ is introduced. By specifying that the shared data streams are conditionally independent on the underlying process $\mathbf{x}$ i.e. $p(\mathbf{f}_1, \mathbf{f}_2 \mid \mathbf{x}) = p(\mathbf{f}_1 \mid \mathbf{x})p(\mathbf{f}_2 \mid \mathbf{x})$ and consequently $p(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}) = p(\mathbf{y}_1 \mid \mathbf{x})p(\mathbf{y}_2 \mid \mathbf{x})$, it is expected that $\mathbf{x}$ will model some shared information between $\mathbf{y}_1$ and $\mathbf{y}_2$, since $\mathbf{x}$ is the only thing the two data sets have in common. The corresponding graphical model is shown in Figure 3.2a.

3.2.1.2 Modelling the private information

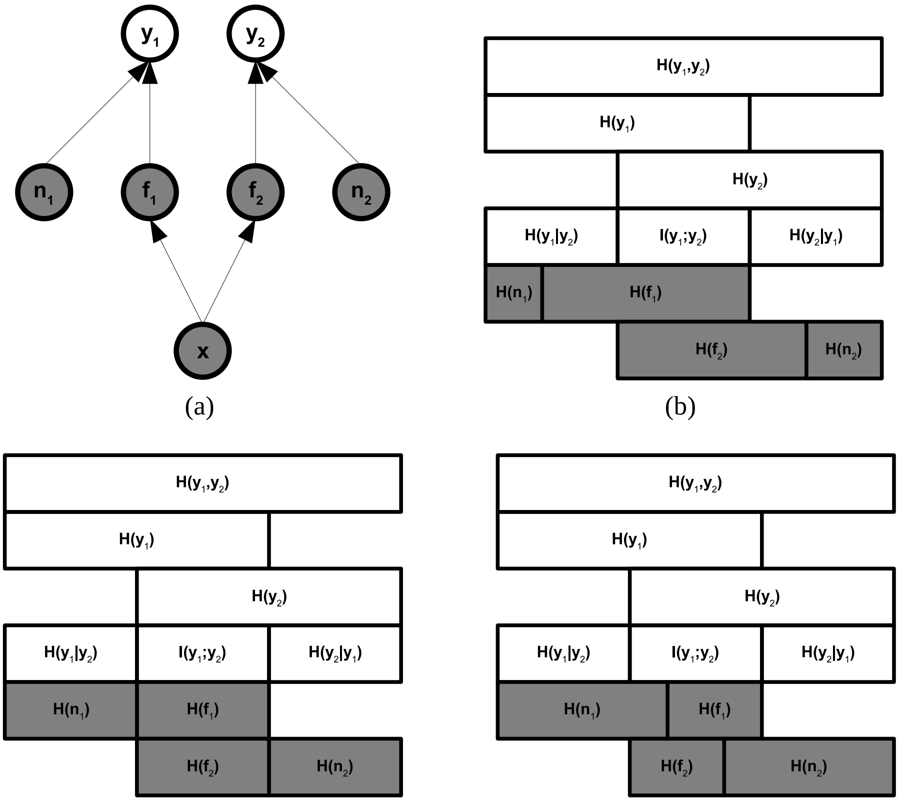
However, we want $\mathbf{x}$ to *only* model the shared information, and not any of the private information contained within each source, so it is necessary to add a further constraint on the model. It is stated in (Klami & Kaski, 2006) that a necessary condition for a generative model to accurately find dependencies between two data sets is for the model to contain enough flexibility to model the marginals $p(\mathbf{y}_1)$ and $p(\mathbf{y}_2)$ with the noise processes. However, we suggest that the model should be able to accurately model $p(\mathbf{y}_1 \mid \mathbf{y}_2)$ and $p(\mathbf{y}_2 \mid \mathbf{y}_1)$ with $p(\mathbf{n}_1)$ and $p(\mathbf{n}_2)$ respectively, following (3.8) and (3.9), such that each noise process is flexible enough to model all of the private information ($H(\mathbf{y}_1 \mid \mathbf{y}_2)$ and $H(\mathbf{y}_2 \mid \mathbf{y}_1)$) contained within each data set. Given that the data density estimated by the model is a good approximation of the true data density we can write:

$$H(\mathbf{y}_1) = H(\mathbf{f}_1) + H(\mathbf{n}_1), H(\mathbf{y}_2) = H(\mathbf{f}_2) + H(\mathbf{n}_2) \tag{3.12}$$

If we maximise the amount of private information from $\mathbf{y}_1$ and $\mathbf{y}_2$ that is captured by $\mathbf{n}_1$ and $\mathbf{n}_2$ respectively, such that $\mathbf{n}_1$ and $\mathbf{n}_2$ capture $H(\mathbf{y}_1 \mid \mathbf{y}_2)$ and $H(\mathbf{y}_2 \mid \mathbf{y}_1)$, the leftover uncertainty in the data will therefore be the shared information between $\mathbf{y}_1$ and $\mathbf{y}_2$ i.e. $H(\mathbf{f}_1) = H(\mathbf{f}_2) = I(\mathbf{y}_1; \mathbf{y}_2)$:

$$H(\mathbf{y}_1) = I(\mathbf{y}_1; \mathbf{y}_2) + H(\mathbf{y}_1 \mid \mathbf{y}_2), H(\mathbf{y}_2) = I(\mathbf{y}_1; \mathbf{y}_2) + H(\mathbf{y}_2 \mid \mathbf{y}_1) \tag{3.13}$$

This concept is illustrated in Figures 3.2(b), (c), and (d) for different constraints on the noise processes $\mathbf{n}_1$ and $\mathbf{n}_2$ for the generative model, whose structure is shown in (a). (b), (c) and (d) show the relationship between the entropies of two correlated variables $\mathbf{y}_1$ and $\mathbf{y}_2$ (white blocks) and the model components $\mathbf{f}_1$, $\mathbf{f}_2$, $\mathbf{n}_1$ and $\mathbf{n}_2$ (grey blocks). In (b) and (c), $\mathbf{n}_1$ and $\mathbf{n}_2$ are constrained to be independent of each other, such that they cannot model any of the shared information $I(\mathbf{y}_1; \mathbf{y}_2)$. In (b) the noise processes are not flexible enough to capture all of the private information contained within each data set, such that the shared components $\mathbf{f}_1$ and $\mathbf{f}_2$ are forced to model some of the private information as well as the shared information $I(\mathbf{y}_1; \mathbf{y}_2)$. In the ideal case, shown in (c), the noise processes are sufficiently flexible such that $\mathbf{n}_1$ and $\mathbf{n}_2$ exactly capture the private information and $I(\mathbf{f}_1; \mathbf{f}_2) = I(\mathbf{y}_1; \mathbf{y}_2)$. In (d) the noise processes are too flexible; such that $\mathbf{n}_1$ and $\mathbf{n}_2$ are free to model some of the shared information $I(\mathbf{y}_1; \mathbf{y}_2)$ and $I(\mathbf{f}_1; \mathbf{f}_2) < I(\mathbf{y}_1; \mathbf{y}_2)$.



(a) (b)

(c) (d)

Figure 3.2: The relationship between joint entropy, marginal entropy, conditional entropy, and mutual information for two correlated variables $\mathbf{y}_1$ and $\mathbf{y}_2$, shown with the entropies for the underlying functions $\mathbf{f}_1$ and $\mathbf{f}_2$, and noise $\mathbf{n}_1$ and $\mathbf{n}_2$ for different configurations of the model shown in (a). In (d), $\mathbf{n}_1$ and $\mathbf{n}_2$ need not be independent. In (b) and (c) $\mathbf{n}_1$ and $\mathbf{n}_2$ are independent; i.e. it is assumed that $p(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}) = p(\mathbf{y}_1 \mid \mathbf{x})p(\mathbf{y}_2 \mid \mathbf{x})$. In (c), the model contains enough flexibility for the noise to maximally model the marginals, such that the underlying functions are forced to model the shared components of $\mathbf{y}_1$ and $\mathbf{y}_2$.

3.3 Linear generative models

In this section, we look at generative dependency seeking models for modelling two Gaussian distributed data variables $\mathbf{y}_1$ and $\mathbf{y}_2$ which have a linear relationship. Finding linearly correlated features between two data sets can be solved by the discriminative method of canonical correlation analysis (CCA) (Hotelling, 1936), which we reviewed in Section 2.5.3. However, CCA does not define a probability density for the data, a problem which has been addressed by its probabilistic formulation in (Bach & Jordan, 2005). Whereas CCA maximises the correlation between the extracted features (termed the canonical variates) from each data set, a generative approach *a priori* models the data sets as having maximally correlated features (i.e. identical features) through a shared underlying latent variable. In the generative model, each data variable is modelled as a sum of a shared component, which is linearly related to an underlying shared latent variable $\mathbf{x}$, and a noise component $\mathbf{n}$. The generative process for the data is given by:

$$\mathbf{y}_1 = \mathbf{W}_1\mathbf{x} + \mathbf{n}_1 \tag{3.14}$$

$$\mathbf{y}_2 = \mathbf{W}_2\mathbf{x} + \mathbf{n}_2 \tag{3.15}$$

where $\mathbf{y}_1 \in \Re^{m_1}, \mathbf{y}_2 \in \Re^{m_2}$ such that each data stream is linearly related to a shared underlying process $\mathbf{x} \in \Re^q$, by the matrices $\mathbf{W}_1 \in \Re^{m_1 \times q}$, $\mathbf{W}_2 \in \Re^{m_2 \times q}$. If we suppose that $\mathbf{x} \sim N(0, \mathbf{I})$, $\mathbf{n_1} \sim N(0, \Psi_1)$ and $\mathbf{n_2} \sim N(0, \Psi_2)$ then we obtain a Gaussian latent variable model as discussed in Section 2.2.1.

Following the discussion in the previous section, we create flexible noise processes by specifying that $\Psi_1 \in \Re^{m_1 \times m_1}$ and $\Psi_2 \in \Re^{m_2 \times m_2}$ are full covariance matrices, so that $p(\mathbf{n}_1)$ and $p(\mathbf{n}_2)$ can approximate $p(\mathbf{y}_1 \mid \mathbf{y}_2)$ and $p(\mathbf{y}_2 \mid \mathbf{y}_1)$ respectively. The resultant generative model is the probabilistic canonical correlation analysis model of Bach and Jordan (Bach & Jordan, 2005). If we had constrained Ψ_1 and Ψ_2, for instance if we had assumed isotropic noise: $\Psi_1 = \Psi_2 = \sigma^2\mathbf{I}$, $\mathbf{x}$ would capture correlations within, as well as between, the two data streams i.e. the resultant model would be probabilistic PCA as reviewed in Section 2.4.1. This scenario is illustrated in Figure 3.2b.

3.3.1 Introducing correlations through linear mixtures

We note that the covariance matrix of the data under the model is given by:

$$\begin{aligned} \mathbf{\Sigma}_y &= E\left(\begin{pmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{pmatrix}\begin{pmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{pmatrix}^\top\right) = \mathbf{\Sigma}_f + \mathbf{\Sigma}_n \qquad (3.16)\\ \text{where } \mathbf{\Sigma}_f &= \begin{pmatrix}\mathbf{W}_1\mathbf{W}_1^\top & \mathbf{W}_1\mathbf{W}_2^\top\\ \mathbf{W}_2\mathbf{W}_1^\top & \mathbf{W}_2\mathbf{W}_2^\top\end{pmatrix}, \mathbf{\Sigma}_n = \begin{pmatrix}\mathbf{\Psi}_1 & 0\\ 0 & \mathbf{\Psi}_2\end{pmatrix}\end{aligned}$$

where $\mathbf{\Sigma}_f$ models variation shared between $\mathbf{y}_1$ and $\mathbf{y}_2$, the between-set variation, and $\mathbf{\Sigma}_n$ models variation that is contained within $\mathbf{y}_1$ and $\mathbf{y}_2$, the within-set variation. Consider the linear transformation of the data

$$\begin{pmatrix}\mathbf{z}_1\\ \mathbf{z}_2\end{pmatrix} = \begin{pmatrix}\mathbf{\Psi}_1^{-\frac{1}{2}} & 0\\ 0 & \mathbf{\Psi}_2^{-\frac{1}{2}}\end{pmatrix}\begin{pmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{pmatrix} = \begin{pmatrix}\mathbf{\Psi}_1^{-\frac{1}{2}}\mathbf{W}_1\mathbf{x} + \mathbf{\Psi}_1^{-\frac{1}{2}}\mathbf{n}_1\\ \mathbf{\Psi}_2^{-\frac{1}{2}}\mathbf{W}_2\mathbf{x} + \mathbf{\Psi}_2^{-\frac{1}{2}}\mathbf{n}_2\end{pmatrix} \qquad (3.17)$$

where $\mathbf{z}_1 \in \Re^{m_1}$, and $\mathbf{z}_2 \in \Re^{m_2}$. Since the sample covariance of the transformed noise components of $\mathbf{z}_1$ and $\mathbf{z}_2$ are $\mathbf{\Psi}_1^{-\frac{1}{2}}E(\mathbf{n}_1\mathbf{n}_1^\top)\mathbf{\Psi}_1^{-\frac{1}{2}} = \mathbf{I}_{m_1}$ and $\mathbf{\Psi}_2^{-\frac{1}{2}}E(\mathbf{n}_2\mathbf{n}_2^\top)\mathbf{\Psi}_2^{-\frac{1}{2}} = \mathbf{I}_{m_2}$ respectively, i.e. isotropic noise with unit variance, it follows that the elements of $\mathbf{z}_1$ are uncorrelated given $\mathbf{x}$, and similarly for $\mathbf{z}_2$. In fact, $\mathbf{z} = [\mathbf{z}_1^\top\mathbf{z}_2^\top]^\top$ is generated according to a probabilistic PCA model with weight matrix $\mathbf{V} = [\mathbf{V}_1^\top\mathbf{V}_2^\top]^\top$ with $\mathbf{V}_1 = \mathbf{\Psi}_1^{-\frac{1}{2}}\mathbf{W}_1, \mathbf{V}_2 = \mathbf{\Psi}_2^{-\frac{1}{2}}\mathbf{W}_2$, and a fixed noise variance of 1. Each data variable may then be written as a linear mixture of independent functions which are all linearly related to a shared latent variable $\mathbf{x}$:

$$\begin{aligned}\mathbf{y}_1 &= \mathbf{\Psi}_1^{\frac{1}{2}}\mathbf{z}_1 = \mathbf{\Psi}_1^{\frac{1}{2}}\mathbf{V}_1\mathbf{x} + \mathbf{n}_1\\ \mathbf{y}_2 &= \mathbf{\Psi}_2^{\frac{1}{2}}\mathbf{z}_2 = \mathbf{\Psi}_2^{\frac{1}{2}}\mathbf{V}_2\mathbf{x} + \mathbf{n}_2 \qquad (3.18)\end{aligned}$$

where $\mathbf{n}_1$ and $\mathbf{n}_2$ are distributed as before. With this interpretation of the probabilistic CCA model, the within-set variation is modelled by a linear transformation of indepen-

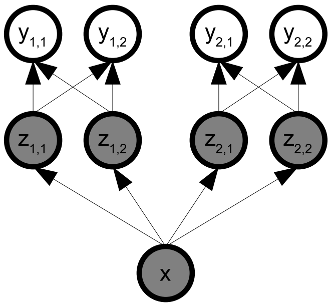


Figure 3.3: Graphical model for a new interpretation of probabilistic canonical correlation analysis. An intermediate set of latent variables $\mathbf{z} = [\mathbf{z}_1^\top \mathbf{z}_2^\top]^\top$ is introduced, where $\mathbf{z}_1 \in \Re^{m_1}, \mathbf{z}_2 \in \Re^{m_2}$ $(m_1 = m_2 = 2)$, which are conditionally independent of a shared latent variable $\mathbf{x} \in \Re^q (q = 1)$, such that $\mathbf{x}$ models the correlations between the elements of $\mathbf{z}$. Each data source is modelled by a linear mixture of the independent underlying functions: $\mathbf{y}_1 = \mathbf{\Psi}_1^{\frac{1}{2}}\mathbf{z}_1, \mathbf{y}_2 = \mathbf{\Psi}_2^{\frac{1}{2}}\mathbf{z}_2$, where $\mathbf{\Psi}_1^{\frac{1}{2}} \in \Re^{m_1 \times m_1}, \mathbf{\Psi}_2^{\frac{1}{2}} \in \Re^{m_2 \times m_2}$, such that the within- set variation is modelled through a linear mixture of independent noise processes. This differs from the original probabilistic CCA model in (Bach & Jordan, 2005) in which the within-set variation is modelled by an additive noise component correlated across the data dimensions.

dent noise processes. We can think of probabilistic canonical correlation analysis as probabilistic principal component analysis on two linearly transformed data variables $\mathbf{z}_1$ and $\mathbf{z}_2$, where the transformations $\mathbf{\Psi}_1^{-1/2}$ and $\mathbf{\Psi_2}^{-1/2}$ remove the within-set variation such that the weight vectors $\mathbf{V}_1$ and $\mathbf{V}_2$ span the between-set variation. This idea is shown graphically in Figure 3.3 for the case of a one dimensional latent variable $\mathbf{x}$ $(q = 1)$ and where each data variable is two dimensional $(m_1 = m_2 = 2)$.

3.3.2 An alternative version of Probabilistic Canonical Correlation Analysis

The latent variable model for the different interpretation of canonical correlation analysis introduced in Section 3.3.1 is given by:

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid 0, \mathbf{I}_q), \qquad min(m_1, m_2) \geq q \geq 1 \tag{3.19}$$

$$p(\mathbf{z}_1 \mid \mathbf{x}, \mathbf{V}_1) = \mathcal{N}(\mathbf{z}_1 \mid \mathbf{V}_1\mathbf{x}, \mathbf{I}_{m_1}), \qquad \mathbf{V}_1 \in \Re^{m_1 \times q} \tag{3.20}$$

$$p(\mathbf{z}_2 \mid \mathbf{x}, \mathbf{V}_2) = \mathcal{N}(\mathbf{z}_2 \mid \mathbf{V}_2\mathbf{x}, \mathbf{I}_{m_2}), \qquad \mathbf{V}_2 \in \Re^{m_2 \times q} \tag{3.21}$$

$$p(\mathbf{y}_1 \mid \mathbf{z}_1, \mathbf{\Psi}_1) = \delta(\mathbf{y}_1 - (\mathbf{\Psi}_1^{1/2}\mathbf{z}_1 + \mu_1)), \quad \mathbf{\Psi}_1 \in \Re^{m_1 \times m_1}, \mu_1 \in \Re^{m_1} \tag{3.22}$$

$$p(\mathbf{y}_2 \mid \mathbf{z}_2, \mathbf{\Psi}_2) = \delta(\mathbf{y}_2 - (\mathbf{\Psi}_2^{1/2}\mathbf{z}_2 + \mu_2)), \quad \mathbf{\Psi}_2 \in \Re^{m_2 \times m_2}, \mu_2 \in \Re^{m_2} \tag{3.23}$$

where we have used μ_1 and μ_2 to allow for a bias term on $\mathbf{y}_1$ and $\mathbf{y}_2$ respectively, and

$$p(\mathbf{y}_1 \mid \mathbf{x}) = \int p(\mathbf{y}_1 \mid \mathbf{z}_1)p(\mathbf{z}_1 \mid \mathbf{x})d\mathbf{z}_1 = \mathcal{N}(\mathbf{y}_1 \mid \mathbf{\Psi}_1^{1/2}\mathbf{V}_1\mathbf{x} + \mu_1, \mathbf{\Psi}_1) \tag{3.24}$$

$$p(\mathbf{y}_2 \mid \mathbf{x}) = \int p(\mathbf{y}_2 \mid \mathbf{z}_2)p(\mathbf{z}_2 \mid \mathbf{x})d\mathbf{z}_2 = \mathcal{N}(\mathbf{y}_2 \mid \mathbf{\Psi}_2^{1/2}\mathbf{V}_2\mathbf{x} + \mu_2, \mathbf{\Psi}_2) \tag{3.25}$$

Again, this model (like the original probabilistic CCA model) is limited since it models the relationship between the latent and data spaces as linear, which may be insufficient for data which lie close to nonlinear manifolds embedded in data space. However, this above formulation of probabilistic CCA can then be extended to modelling nonlinear relationships, as seen in the following section. The standard approach for fitting this latent variable model is to marginalise the latent variables $\mathbf{x}$, and to optimise the parameters $\mathbf{V} = [\mathbf{V}_1^\top, \mathbf{V}_2^\top]^\top$, $\mathbf{\Psi}_1^{1/2}$ and $\mathbf{\Psi}_2^{1/2}$ via maximum likelihood. We follow the dual approach, used in the derivation of Gaussian Process Latent Variable Models (Lawrence, 2004, 2005), which is to marginalise the parameters and to optimise the likelihood with respect to the latent variables.

3.4 A GPLVM version of CCA

Gaussian process latent variable models (GPLVM) described in (Lawrence, 2004, 2005) are a new class of probabilistic models that define Gaussian process ‘mappings’ from a latent space to the data space. A theoretical grounding is provided for the GPLVM, deriving the model from a dual formulation of probabilistic principal component analysis (PPCA) (Tipping & Bishop, 1999). Rather than integrating out the latent variables and optmising the linear mapping of the PPCA model as in (Tipping & Bishop, 1999), the GPLVM approach is to integrate out the mapping and optimise the latent variable positions. The resulting model is a product of D independent Gaussian processes (where D is the dimension of the data), where the process inputs are the latent variables. PPCA is a special case of the GPLVM when the model’s covariance function is linear, but any valid covariance function can be used, such that there is an implicit nonlinear mapping from the latent space to the data space, such that the GPLVM is a probabilistic model of nonlinear principal component analysis.

In the GPLVM, the data (output) dimensions are *a priori* assumed to be independent and identically distributed, such that the latent coordinates, which are common to all dimensions, capture the variation between the dimensions. This model is therefore not appropriate for capturing variations between two related data sets, as we noted in Section 3.3, since the model's set of latent coordinates will capture the private information (or within-set variation) as well as the shared information. One approach in the literature to finding structure between two data sets is to optimise two GPLVM's that have a joint latent space (Shon *et al.*, 2006). This relaxes the 'identically distributed' constraint on the data dimensions of the GPLVM - each data set is modelled by a GPLVM which has its own covariance function. However, we argue that this is not strictly a dependency seeking model, since the private information within each data set is not explicitly modelled.

Our approach to creating a dependency seeking generative model is to model the within-set variation by using a linear combination of underlying Gaussian processes with a common input, generalising from the new interpretation of probabilistic canonical correlation analysis introduced in Section 3.3.2. This is equivalent to relaxing the 'independently distributed' assumption on the data dimensions within each data set; a GPLVM underlies each data set, and the output dimensions are linearly mixed to model dependencies within each data set. We describe the model in the next section.

3.4.1 Derivation of the model

Starting from the new interpretation of probabilistic canonical correlation analysis in Section 3.3.2, the set of N data pairs $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2]$ (where $\mathbf{Y}_1 = [\mathbf{y}_{1,1}, ..., \mathbf{y}_{1,N}]^\top$ and $\mathbf{Y}_2 = [\mathbf{y}_{2,1}, ..., \mathbf{y}_{2,N}]^\top$) is modelled as a linear combination of a set of underlying function values, $\mathbf{Z} = [\mathbf{Z}_1, \mathbf{Z}_2]$, where $\mathbf{Z}_1 = [\mathbf{z}_{1,1}, ..., \mathbf{z}_{1,N}]^\top$ and $\mathbf{Z}_2 = [\mathbf{z}_{2,1}, ..., \mathbf{z}_{2,N}]^\top$:

$$p(\mathbf{Y} \mid \mathbf{Z}, \mathbf{\Psi}) = \prod_{i=1}^{m_1+m_2} \delta(\mathbf{Y}(:,i) - (\mathbf{Z}\mathbf{\Psi}(:,i)^{\frac{1}{2}})) \tag{3.26}$$

where we have assumed zero mean data, $\mathbf{Y}(:,i)$ is the ith column of $\mathbf{Y}$, and $\mathbf{\Psi}(:,i)^{\frac{1}{2}}$ is the ith column of $\mathbf{\Psi}^{\frac{1}{2}} = \begin{pmatrix} \mathbf{\Psi}_1^{\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Psi}_2^{\frac{1}{2}} \end{pmatrix}$. In our PCCA model the prior on the latent function values $\mathbf{Z} = [\mathbf{Z}_1, \mathbf{Z}_2]$ is given by:

$$p(\mathbf{Z} \mid \mathbf{X}, \mathbf{V}) = \prod_{i=1}^{m_1+m_2} \mathcal{N}(\mathbf{Z}(:,i) \mid \mathbf{X}\mathbf{v}_i^\top, \mathbf{I}_N) \tag{3.27}$$

where $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]^\top$ is the set of N latent variables underlying $\mathbf{Z}$, $\mathbf{Z}(:,i)$ is the ith column of $\mathbf{Z}$, and $\mathbf{v}_i$ is the ith row of $\mathbf{V}$. We can think of each latent function value $\mathbf{z}_n = [\mathbf{z}_{1,n}^\top \mathbf{z}_{2,n}^\top]^\top$ as being a function of $\mathbf{x}_n$ such that the columns of $\mathbf{Z}$ are the latent common functions evaluated at $\mathbf{X}$. In (3.27), the latent functions are linear functions of their inputs, but we see in Section 3.5 that we can also consider nonlinear functions with the model.

3.4.1.1 Integrating out the linear mapping

Following the derivation of the GPLVM in (Lawrence, 2004), a prior conjugate to (3.27) is placed on $\mathbf{V}$ (which parameterises the mapping from $\mathbf{X}$ to $\mathbf{Z}$), and then we integrate over $\mathbf{V}$. An isotropic Gaussian prior with unit variance is used:

$$p(\mathbf{V}) = \prod_{i=1}^{m_1+m_2} \mathcal{N}(\mathbf{v}_i \mid \mathbf{0}, \mathbf{I}_{m_1+m_2}) \tag{3.28}$$

where $\mathbf{v}_i$ is the ith row of $\mathbf{V}$. The resulting marginal likelihood is given by:

$$\begin{aligned} p(\mathbf{Z} \mid \mathbf{X}) &= \int p(\mathbf{Z} \mid \mathbf{X}, \mathbf{V}) p(\mathbf{V}) d\mathbf{V} && (3.29) \\ &= \prod_{i=1}^{m_1+m_2} \mathcal{N}(\mathbf{Z}(:,i) \mid \mathbf{0}, \mathbf{X}\mathbf{X}^\top + \mathbf{I}_N) && (3.30) \\ &= \frac{1}{(2\pi)^{\frac{DN}{2}} |\mathbf{K}|^{\frac{D}{2}}} \text{exp} \left(-\frac{1}{2} \text{tr}(\mathbf{K}^{-1}\mathbf{Z}\mathbf{Z}^\top) \right) && (3.31) \end{aligned}$$

where $\mathbf{K} = \mathbf{X}\mathbf{X}^\top + \mathbf{I}_N$ and $D = m_1 + m_2$. This is a GPLVM, consisting of a product of $m_1 + m_2$ independent Gaussian processes. The nth value for each data source is a linear combination of the nth latent function values evaluated at $\mathbf{x}_n$ as in (3.22) and

(3.23). The likelihood function for $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2]$ is given by integrating out $\mathbf{Z}$, where we have used (3.26) and the prior on $\mathbf{Z}$ in (3.31):

$$\begin{aligned} p(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\Psi}) &= \int p(\mathbf{Y} \mid \mathbf{Z}, \boldsymbol{\Psi}) p(\mathbf{Z} \mid \mathbf{X}) d\mathbf{Z} \\ &= \frac{1}{(2\pi)^{\frac{DN}{2}} |\mathbf{K}|^{\frac{D}{2}} |\boldsymbol{\Psi}|^{\frac{N}{2}}} \text{exp} \left(-\frac{1}{2} \text{tr}(\mathbf{K}^{-1} \mathbf{Y} \boldsymbol{\Psi}^{-1} \mathbf{Y}^{\top}) \right) \end{aligned} \tag{3.32}$$

The log likelihood function for the model is given by:

$$\mathcal{L}_{\mathbf{Y}|\mathbf{X}} = -\frac{N}{2} \text{ln} |\boldsymbol{\Psi}| - \frac{DN}{2} \text{ln}(2\pi) - \frac{D}{2} \text{ln} |\mathbf{K}| - \frac{1}{2} \text{tr}(\mathbf{K}^{-1} \mathbf{Y} \boldsymbol{\Psi}^{-1} \mathbf{Y}^{\top}) \tag{3.33}$$

The gradients of (3.33) with respect to $\mathbf{X}$ is given by:

$$\frac{\partial \mathcal{L}_{\mathbf{Y}|\mathbf{X}}}{\partial \mathbf{X}} = -\frac{D}{2} \mathbf{K}^{-1} \mathbf{X} + \frac{1}{2} \mathbf{K}^{-1} \mathbf{Y} \boldsymbol{\Psi}^{-1} \mathbf{Y}^{\top} \mathbf{K}^{-1} \mathbf{X} \tag{3.34}$$

and a fixed point where the gradients are zero is given by:

$$\frac{1}{D} \mathbf{Y} \boldsymbol{\Psi}^{-1} \mathbf{Y}^{\top} \mathbf{K}^{-1} \mathbf{X} = \mathbf{X} \tag{3.35}$$

which is satisfied by:

$$\mathbf{X} = \mathbf{U}_q \mathbf{L}_q \mathbf{R}^{\top} \tag{3.36}$$

where $\mathbf{U}_q$ are the q dominant eigenvectors of $\mathbf{Y} \boldsymbol{\Psi}^{-1} \mathbf{Y}^{\top}$, $\mathbf{L}_q$ is the diagonal matrix $(\boldsymbol{\Lambda}_q - \mathbf{I}_q)^{\frac{1}{2}}$ with $\boldsymbol{\Lambda}_q$ being the corresponding diagonal matrix of eigenvalues, and $\mathbf{R} \in \Re^{q \times q}$ is a rotation matrix. The gradients of (3.33) with respect to $\boldsymbol{\Psi}$ is given by:

$$\frac{\partial \mathcal{L}_{\mathbf{Y}|\mathbf{X}}}{\partial \boldsymbol{\Psi}} = -\frac{N}{2} \boldsymbol{\Psi}^{-1} + \frac{1}{2} \boldsymbol{\Psi}^{-1} \mathbf{Y}^{\top} \mathbf{K}^{-1} \mathbf{Y} \boldsymbol{\Psi}^{-1} \tag{3.37}$$

and a fixed point where the gradients are zero is given by:

$$\boldsymbol{\Psi} = \frac{1}{N} (\mathbf{Y}^{\top} \mathbf{K}^{-1} \mathbf{Y}) \tag{3.38}$$

which we then constrain to be of block diagonal form to give $\boldsymbol{\Psi} = \begin{pmatrix} \boldsymbol{\Psi}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Psi}_2 \end{pmatrix}$ where $\boldsymbol{\Psi}_1 = \mathbf{Y}_1^\top \mathbf{K}^{-1} \mathbf{Y}_1 / N \in \Re^{m_1 \times m_1}$, and $\boldsymbol{\Psi}_2 = \mathbf{Y}_2^\top \mathbf{K}^{-1} \mathbf{Y}_2 / N \in \Re^{m_2 \times m_2}$. $\boldsymbol{\Psi}$ has the interpretation of being the noise covariance matrix of the probabilistic CCA model. $\boldsymbol{\Psi}_1$ models the within-set variation in $\mathbf{Y}_1$, and $\boldsymbol{\Psi}_2$ models the within-set variation in $\mathbf{Y}_2$.

3.4.2 Finding latent coordinates for each data set

After training the model by finding $\mathbf{X}$ and $\boldsymbol{\Psi}$ according to the update equations, we may want to find the latent space representation of just one of the data sets. Denoting the data and optimised latent coordinates as $\mathcal{D} = \{\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{X}\}$, the resulting probability distribution over a data point $\mathbf{y}_{1,n}$ from the first data set given a latent point $\mathbf{x}_n$ is given by:

$$p(\mathbf{y}_{1,n} \mid \mathbf{x}_n, \mathcal{D}) = \mathcal{N}(\mathbf{y}_{1,n} \mid \mu_1(\mathbf{x}_n), \sigma_1^2(\mathbf{x}_n) \boldsymbol{\Psi}_1) \tag{3.39}$$

where

$$\mu_1(\mathbf{x}_n) = [\mathbf{k}(\mathbf{x}_n)^\top \mathbf{K}^{-1} \mathbf{Y}_1]^\top \tag{3.40}$$

$$\sigma_1^2(\mathbf{x}_n) = k - \mathbf{k}(\mathbf{x}_n)^\top \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}_n) \tag{3.41}$$

and $\mathbf{K} = C(\mathbf{X}, \mathbf{X}) + \beta^{-1}\mathbf{I}$, $\mathbf{k}(\mathbf{x}_n) = [C(\mathbf{x}_n, \mathbf{x}_1), ..., C(\mathbf{x}_n, \mathbf{x}_N)]^\top$, $k = C(\mathbf{x}_n, \mathbf{x}_n)$, so that, for the linear covariance function, $C(\mathbf{x}_n, \mathbf{x}_n) = \mathbf{x}_n^\top \mathbf{x}_n$. Similarly, the resulting probability distribution over a data point $\mathbf{y}_{2,n}$ from the second data set given a latent point $\mathbf{x}_n$ is given by:

$$p(\mathbf{y}_{2,n} \mid \mathbf{x}_n, \mathcal{D}) = \mathcal{N}(\mathbf{y}_{2,n} \mid \mu_2(\mathbf{x}_n), \sigma_2^2(\mathbf{x}_n) \boldsymbol{\Psi}_2) \tag{3.42}$$

where

$$\mu_2(\mathbf{x}_n) = [\mathbf{k}(\mathbf{x}_n)^\top \mathbf{K}^{-1}\mathbf{Y}_2]^\top \tag{3.43}$$

$$\sigma_2^2(\mathbf{x}_n) = k - \mathbf{k}(\mathbf{x}_n)^\top \mathbf{K}^{-1}\mathbf{k}(\mathbf{x}_n) \tag{3.44}$$

The latent coordinates for a pair of data points is found through:

$$\mathbf{x}_1 = \arg\max_{\mathbf{x}} \ \ln p(\mathbf{y}_1 \mid \mathbf{x}, \mathcal{D}) \tag{3.45}$$

$$\mathbf{x}_2 = \arg\max_{\mathbf{x}} \ \ln p(\mathbf{y}_2 \mid \mathbf{x}, \mathcal{D}) \tag{3.46}$$

where

$$\ln p(\mathbf{y}_1 \mid \mathbf{x}, \mathcal{D}) = -\frac{m_1}{2}\ln(2\pi) - \frac{1}{2}\ln\sigma_1^2(\mathbf{x}) - \frac{1}{2\sigma_1^2(\mathbf{x})}||\mathbf{\Psi}_1^{-\frac{1}{2}}(\mathbf{y}_1 - \mu_1(\mathbf{x}))||^2 \tag{3.47}$$

$$\frac{\partial \ln p(\mathbf{y}_1 \mid \mathbf{x}, \mathcal{D})}{\partial \mathbf{x}} = -\frac{1}{2}\sigma_1^{-2}(\mathbf{x})\frac{\partial \sigma_1^2(\mathbf{x})}{\partial \mathbf{x}}\left(1 - \frac{||\mathbf{\Psi}_1^{-\frac{1}{2}}(\mathbf{y}_1 - \mu_1(\mathbf{x}))||^2}{(\sigma_1^2(\mathbf{x}))}\right) + \frac{(\mathbf{\Psi}_1^{-\frac{1}{2}}(\mathbf{y}_1 - \mu_1(\mathbf{x})))^\top}{(\sigma_1^2(\mathbf{x}))}\frac{\partial \mu_1(\mathbf{x})}{\partial \mathbf{x}} \tag{3.48}$$

where $\frac{\partial \sigma_1^2(\mathbf{x})}{\partial \mathbf{x}}$ and $\frac{\partial \mu_1(\mathbf{x})}{\partial \mathbf{x}}$ depend on the form of the covariance function C, and similarly for $\ln p(\mathbf{y}_2 \mid \mathbf{x}, \mathcal{D})$.

The probability distributions over the data sets $\mathbf{Y}_1$ and $\mathbf{Y}_2$ given the trained model are given by $p(\mathbf{Y}_1 \mid \mathbf{X}, \mathcal{D}) = \prod_{n=1}^{N} p(\mathbf{y}_{1,n} \mid \mathbf{x}_n, \mathcal{D})$ and $p(\mathbf{Y}_2 \mid \mathbf{X}, \mathcal{D}) = \prod_{n=1}^{N} p(\mathbf{y}_{2,n} \mid \mathbf{x}_n, \mathcal{D})$ using (3.39) and (3.42) respectively. Now we can consider the situation in which we have a trained mapping and we wish to predict one data set from the other. We can denote the latent coordinate sets underlying $\mathbf{Y}_1$ and $\mathbf{Y}_2$ as $\hat{\mathbf{X}}_1$ and $\hat{\mathbf{X}}_2$ respectively, and we find $\hat{\mathbf{X}}_1$ and $\hat{\mathbf{X}}_2$ as:

$$\hat{\mathbf{X}}_1 = \arg\max_{\mathbf{X}} \ \ln p(\mathbf{Y}_1 \mid \mathbf{X}, \mathcal{D}) \tag{3.49}$$

$$\hat{\mathbf{X}}_2 = \arg\max_{\mathbf{X}} \ \ln p(\mathbf{Y}_2 \mid \mathbf{X}, \mathcal{D}) \tag{3.50}$$

Intuitively $\hat{\mathbf{X}}_1$ and $\hat{\mathbf{X}}_2$ represent the most highly correlated portions of $\mathbf{Y}_1$ and $\mathbf{Y}_2$. That is, the best prediction of $\mathbf{Y}_1$ and $\mathbf{Y}_2$ is given by the underlying latent coordinates $\hat{\mathbf{X}}_1$ and $\hat{\mathbf{X}}_2$ which are themselves highly correlated. We can use this fact for prediction.

3.4.3 Prediction of one data set given the other

We want to predict $\mathbf{Y}_2^*$ given new values of the first dataset $\mathbf{Y}_1^*$. Our method consists in finding corresponding latent coordinates $\mathbf{X}^*$ for $\mathbf{Y}_1^*$ using (3.45) and (3.50) and the relationship between $\hat{\mathbf{X}}_1$ and $\hat{\mathbf{X}}_2$ i.e. equating $\hat{\mathbf{X}}_1$ and $\hat{\mathbf{X}}_2$, and then using the coordinates to predict the other data set $\mathbf{Y}_2^*$. The predictive distribution over the second data variable $\mathbf{y}_2^*$ given its corresponding latent coordinate $\mathbf{x}^*$ is given by:

$$p(\mathbf{y}_2^*|\mathbf{x}^*, \mathcal{D}) = \mathcal{N}(\mathbf{y}_2^*|\mu_2(\mathbf{x}^*), \sigma_2^2(\mathbf{x}^*)\mathbf{\Psi}_2) \tag{3.51}$$

where

$$\mu_2(\mathbf{x}^*) = [\mathbf{k}(\mathbf{x}^*)^\top \mathbf{K}^{-1}\mathbf{Y}_2]^\top \tag{3.52}$$

$$\sigma_2^2(\mathbf{x}^*) = k - \mathbf{k}(\mathbf{x}^*)^\top \mathbf{K}^{-1}\mathbf{k}(\mathbf{x}^*) = \sigma_1^2(\mathbf{x}^*) \tag{3.53}$$

and $\mathbf{K} = C(\mathbf{X}, \mathbf{X}) + \beta^{-1}\mathbf{I}$, $\mathbf{k}(\mathbf{x}^*) = [C(\mathbf{x}^*, \mathbf{x}_1), ..., C(\mathbf{x}^*, \mathbf{x}_N)]^\top$, $k = C(\mathbf{x}^*, \mathbf{x}^*)$. We independently optimise the likelihood of each $\mathbf{y}_{1,n}^*$ by finding the corresponding $\mathbf{x}_n^*$, which we use to calculate the predictive distribution for $\mathbf{y}_{2,n}^*$. We can similarly find a predictive distribution for $\mathbf{Y}_1^*$ given $\mathbf{Y}_2^*$.

3.5 Extension to nonlinear processes

The previous sections show how probabilistic CCA can be derived in terms of a GPLVM with a linear covariance function (i.e. dual Probabilistic PCA) on two linearly transformed data sets. We can consider nonlinear covariance functions to allow for nonlinear processes such that the resultant model is a nonlinear version of probabilistic CCA. Due to the nonlinear relationship between the latent space and data space, the resultant model will not be optimisable by an eigenvalue problem.

In the following sections we show how to train the model, following the approach used in the training of Lawrence's GPLVM (Lawrence, 2005). Our implementation of the model is based on Neil Lawrence's GPLVM code available online at `http://www.dcs.shef.ac.uk/~neil/gplvm`. Covariance functions that we will use in this thesis are:

3.5.0.1 Squared exponential covariance function

The squared exponential (SE) or RBF is probably the most widely used kernel in the kernel machines field. It favours smooth functions (since it is infinitely differentiable) whose values fall away to almost zero in regions where there is no data, and has the form:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \alpha \exp\left(-\frac{\gamma}{2}(\mathbf{x}_i - \mathbf{x}_j)^\top(\mathbf{x}_1 - \mathbf{x}_j)\right) + \beta^{-1}\delta_{i,j} \tag{3.54}$$

with hyperparameters $\Theta_{K_{SE}} = \{\alpha, \beta, \gamma\}$, where α is a parameter that controls the scale of the output functions, β is the inverse noise variance, and γ controls the characteristic length scale of the functions.

3.5.0.2 Linear covariance function

The linear covariance function (which we have used earlier) is a matrix of inner products of $\mathbf{X}$ such that the output functions are linearly related to $\mathbf{X}$, and is given by:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \alpha \mathbf{x}_i^\top \mathbf{x}_j + \beta^{-1}\delta_{i,j} \tag{3.55}$$

with hyperparameters $\Theta_{K_{lin}} = \{\alpha, \beta\}$, where α is a scale parameter and β is the inverse noise variance.

3.5.0.3 Polynomial covariance function

The polynomial covariance function is given by:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \alpha \left(w\mathbf{x}_i^\top \mathbf{x}_j + \gamma\right)^d + \beta^{-1}\delta_{i,j} \tag{3.56}$$

with hyperparameters $\Theta_{K_{poly}} = \{\alpha, \beta, \gamma, d, w\}$, where α is a scale parameter, β is the inverse noise variance, d defines the degree of the polynomial, w controls the scale of the dot product component, and γ is a bias parameter.

3.5.1 Using different processes for the data sets

To extend the model, we can use different covariance functions $\mathbf{K}_1$ and $\mathbf{K}_2$ for the processes $\mathbf{Z}_1$ and $\mathbf{Z}_2$ respectively, underlying the data sets. We write the log likelihood function as:

$$\mathcal{L}_{\mathbf{Y}|\mathbf{X}} = \mathcal{L}_{\mathbf{Y}_1|\mathbf{X}} + \mathcal{L}_{\mathbf{Y}_2|\mathbf{X}} \tag{3.57}$$

where

$$\begin{aligned}
\mathcal{L}_{\mathbf{Y}_1|\mathbf{X}} &= \ln p(\mathbf{Y}_1 \mid \mathbf{X}) \\
&= -\frac{N}{2}\ln|\mathbf{\Psi}_1| - \frac{m_1 N}{2}\ln(2\pi) - \frac{m_1}{2}\ln|\mathbf{K}_1| - \frac{1}{2}\text{tr}(\mathbf{K}_1^{-1}\mathbf{Y}_1\mathbf{\Psi}_1^{-1}\mathbf{Y}_1^{\top}) \\
\mathcal{L}_{\mathbf{Y}_2|\mathbf{X}} &= \ln p(\mathbf{Y}_2 \mid \mathbf{X}) \\
&= -\frac{N}{2}\ln|\mathbf{\Psi}_2| - \frac{m_2 N}{2}\ln(2\pi) - \frac{m_2}{2}\ln|\mathbf{K}_2| - \frac{1}{2}\text{tr}(\mathbf{K}_2^{-1}\mathbf{Y}_2\mathbf{\Psi}_2^{-1}\mathbf{Y}_2^{\top})
\end{aligned} \tag{3.58}$$

The model consists of two GPLVM's which share the same set of latent coordinates, and each models a linear transformation of its respective data set.

3.5.2 Training the model

To train the model, we have to find the latent coordinates $\mathbf{X}$, the parameters of the covariance functions $\mathbf{\Theta}_{K_i}, i = 1, 2$, and the linear transformations $\mathbf{\Psi}_1$ and $\mathbf{\Psi}_2$ such that the log likelihood function $\mathcal{L}$ is maximised. Since $\mathcal{L}$ is a highly nonlinear function of $\mathbf{X}$ and $\mathbf{\Theta}_{K_i}, i = 1, 2$, we have to use gradient based optimisation procedures. In our experiments we use scaled conjugate gradients (SCG).

3.5.2.1 Optimisation of the latent points $\mathbf{X}$

The gradients of $\mathcal{L}_{\mathbf{Y}|\mathbf{X}}$ with respect to $\mathbf{K}_i$ is given by:

$$\frac{\partial \mathcal{L}_{\mathbf{Y}_i|\mathbf{X}}}{\partial \mathbf{K}_i} = -\frac{D}{2}\mathbf{K}_i^{-1} + \frac{1}{2}\mathbf{K}_i^{-1}\mathbf{Y}_i\mathbf{\Psi}_i^{-1}\mathbf{Y}_i^{\top}\mathbf{K}_i^{-1} \tag{3.59}$$

The gradients of $\mathcal{L}_{\mathbf{Y}|\mathbf{X}}$ with respect to $\mathbf{X}$ can be obtained by combining (3.59) with $\frac{\partial \mathbf{K}_i}{\partial \mathbf{X}}, i = 1, 2$ using the chain rule, where $\frac{\partial \mathbf{K}_i}{\partial \mathbf{X}}$ depends on the form of the covariance function $\mathbf{K}_i$. Using nonlinear covariance functions introduces more flexibility into the model and rather than seeking a maximum likelihood solution for $\mathbf{X}$ it may be preferable to seek a MAP solution. In our experiments we use a Gaussian prior over $\mathbf{X}$: $p(\mathbf{X}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n \mid \mathbf{0}, \mathbf{I})$ and find a MAP solution for $\mathbf{X}$ by maximising $\mathcal{L}_{\mathbf{Y},\mathbf{X}} = \mathcal{L}_{\mathbf{Y}|\mathbf{X}} + \text{ln}p(\mathbf{X})$ with respect to $\mathbf{X}$, where $\mathcal{L}_{\mathbf{Y}|\mathbf{X}}$ is given in 3.57. This is equivalent to penalising $\mathcal{L}_{\mathbf{Y}|\mathbf{X}}$ with the sum of squared elements of $\mathbf{X}$.

3.5.2.2 Optimisation of $\mathbf{\Theta}_{K_i}$, $i = 1, 2$

The gradients of $\mathcal{L}_{\mathbf{Y}|\mathbf{X}}$ with respect to the covariance function parameters $\mathbf{\Theta}_{K_i}$ are given by combining (3.59) with $\frac{\partial \mathbf{K}_i}{\partial \mathbf{\Theta}_{K_i}}$ using the chain rule. The parameters $\mathbf{\Theta}_{K_i}$ that we work with should be positive so in our experiments we optimise $\mathbf{\Theta}_{K_i}$ in a transformed space by using the transformation $\theta = \text{ln}(1 + \exp(\theta'))$ As before for $\mathbf{X}$, we can seek MAP solutions for $\mathbf{\Theta}_{K_i}$ by first specifying priors over $\mathbf{\Theta}_{K_i}$.

3.5.2.3 Optimisation of $\mathbf{\Psi}$

The parameter $\mathbf{\Psi}$ is found through an exact update as before.

$$\mathbf{\Psi}_1 = \frac{1}{N}(\mathbf{Y}_1^{\top}\mathbf{K}_1^{-1}\mathbf{Y}_1) \tag{3.60}$$

$$\mathbf{\Psi}_2 = \frac{1}{N}(\mathbf{Y}_2^{\top}\mathbf{K}_2^{-1}\mathbf{Y}_2) \tag{3.61}$$

In our experiments we update $\mathbf{\Psi}$ every 5 iterations.

3.5.3 Relation to other models

Our model is closely related to the Gaussian Process Latent Variable Model (Lawrence, 2004, 2005) of Lawrence as we reviewed in 2.4.3. Lawrence's model is derived from a dual approach to probabilistic PCA, assuming *a priori* that the data dimensions are independent and identically distributed given the latent variables. The marginal likelihood of the resultant model is a product of D independent Gaussian processes (where D is the dimensionality of the data), and each dimension is identically distributed i.e. they share the same covariance function. The latent coordinates $\mathbf{X}$ are the inputs to the Gaussian processes and are 'mapped' to a distribution over each data dimension. Our model is designed to find relationships between two data sets $\mathbf{Y}_1$ and $\mathbf{Y}_2$ and is derived from a dual approach to probabilistic CCA. The data in the individual dimensions of each data set are assumed to be dependent on each other but independent of the data from the dimensions of the other set, given the shared latent variable set $\mathbf{X}$. The data sets $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are modelled as linear mixtures of independent Gaussian processes $\mathbf{Z}_1$ and $\mathbf{Z}_2$ respectively, which share the same covariance function and the same input set $\mathbf{X}$. An interpretation of the model is that $\mathbf{Z} = [\mathbf{Z}_1, \mathbf{Z}_2]$, linear transformations of the data sets, are generated according to a GPLVM.

The Scaled Gaussian Process Latent Variable Model (SGPLVM) of Grochow et al. (Grochow *et al.*, 2004) is an extension of the GPLVM and associates a scale parameter with each dimension of the data. The likelihood function for this model is given by:

$$p(\mathbf{Y} \mid \mathbf{X}, \mathbf{W}) = \frac{|\mathbf{W}|^{\frac{N}{2}}}{(2\pi)^{\frac{DN}{2}}|\mathbf{K}|^{\frac{D}{2}}} \exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{K}^{-1}\mathbf{Y}\mathbf{W}\mathbf{Y}^{\top})\right) \tag{3.62}$$

for a D-dimensional data set $\mathbf{Y}$, latent variable set $\mathbf{X}$, and the diagonal matrix $\mathbf{W} \in \Re^{D\times D}$ of scale parameters $\{w_1, ..., w_D\}$. The model is similar to Factor Analysis, in that the data dimensions are assumed to be independent, but allowed to have different noise variances i.e. the dimensions are not identically distributed. From (3.62) it can be seen that the distribution over the dth data dimension is a Gaussian process with a covariance function $w_d^{-1}\mathbf{K}$. Since the different noise variances of the data dimensions are already accounted for by the model, $\mathbf{X}$ captures the correlations between the data dimensions. For our model, the likelihood function is given by:

$$p(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\Psi}) \;=\; \frac{1}{(2\pi)^{\frac{DN}{2}}|\mathbf{K}|^{\frac{D}{2}}|\boldsymbol{\Psi}|^{\frac{N}{2}}}\exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{K}^{-1}\mathbf{Y}\boldsymbol{\Psi}^{-1}\mathbf{Y}^{\top})\right) \tag{3.63}$$

for two data sets $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2]$, and the block diagonal matrix of parameters $\boldsymbol{\Psi} = \begin{pmatrix} \boldsymbol{\Psi}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Psi}_2 \end{pmatrix}$ where $\boldsymbol{\Psi}_1 \in \Re^{m_1 \times m_1}$, and $\boldsymbol{\Psi}_2 \in \Re^{m_2 \times m_2}$, where m_1 and m_2 are the dimensions of $\mathbf{Y}_1$ and $\mathbf{Y}_2$ respectively. From (3.63) it can be seen that the covariance function between the ith and jth dimensions of $\mathbf{Y}$ is $\left(\boldsymbol{\Psi}^{-1}_{i,j}\right)^{-1}\mathbf{K}$ where $\boldsymbol{\Psi}^{-1}_{i,j}$ is the (i,j)th element of the matrix $\boldsymbol{\Psi}^{-1}$. Due to the block diagonal structure of $\boldsymbol{\Psi}^{-1}$, there are cross covariance functions between the variables within each data set. By accounting for the correlations within each data set with the model, $\mathbf{X}$ should capture the between-set variation.

Our model, like the SGPLVM, can be interpreted as a warped Gaussian processes (Snelson *et al.*, 2004) with a linear warping function $\mathbf{z}_n = \boldsymbol{\Psi}^{-\frac{1}{2}}\mathbf{y}_n$. In the warped Gaussian process framework, a transformation is made from the data space to a latent space, such that the data is best modelled by a Gaussian process in the latent space. Rather than being an ad-hoc step, this preprocessing transformation is found automatically since it is incorporated into the probabilistic framework of the GP. A warped Gaussian process is defined as follows. The latent function values $\mathbf{Z} = [z_1, ..., z_N]^{\top}$ is modelled by a Gaussian process with zero mean and covariance function $\mathbf{K}$, parameterised by Θ, and the transformation from the data space to the latent function space is given by a mapping of the N data points $\mathbf{Y} = [y_1, ..., y_N]^{\top}$ through the same function f

$$z_n = f(y_n, \boldsymbol{\Psi}) \tag{3.64}$$

where f is required to be monotonic and maps to the whole of the real line, such that probability measure is conserved in the transform, and $\boldsymbol{\Psi}$ parameterises the transform.

The log likelihood function ln$p(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\Psi}, \Theta)$ is given by:

$$\mathcal{L} = -\frac{1}{2}\sum_{n=1}^{N} \ln \left.\frac{\partial f(y, \boldsymbol{\Psi})}{\partial y}\right|_{y_n} - \frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln|\mathbf{K}| - \frac{1}{2}\mathrm{tr}(\mathbf{K}^{-1} f(\mathbf{Y}) f(\mathbf{Y})^{\top}) \quad (3.65)$$

where the first term is the Jacobian term that takes the transformation into account. The warped GP is a generalisation of the standard GP, finding extra structure in the (possibly non Gaussian) data by learning a transformation of the data. In (Snelson *et al.*, 2004), the analysis is limited to one dimensional regression problems, but in our case, the learned transformation from the data to the latent function space models the within-set variation between two data sets such that the variation between the data sets is best modelled by a GPLVM.

In our model, using a linear mixture of Gaussian processes to model correlations within a data set is similar in spirit to the Semiparametric Latent Factor Model (Teh *et al.*, 2005). This is a semiparametric model for regression problems involving multiple response variables. The model uses a linear mixture of Gaussian processes to capture the dependencies that may exist between the response variables.

3.6 Experiments

In this section, we demonstrate the GPLVM-CCA model on a range of data sets. In Section 3.6.1, we present results on a pair of data sets which have an underlying linear relationship. In Section 3.6.2, we illustrate the algorithm on a nonlinear CCA problem. In Section 3.6.4 we demonstrate the GPLVM-CCA on a set of image pairs. Each pair consists of the left and right half of a face with varying poses and expressions. We find a joint latent space for the data, and show how to predict one face half given the other. Additionally, in Section 3.6.5, we show how the we can still predict a face half given the other face half that has pixel values missing at random.

3.6.1 Linear example

We demonstrate the GPLVM-CCA model on a simple toy problem to show the ability of the model to find correlated features between two data sets. We create 200 data pairs according to

$$\begin{aligned}\mathbf{y}_1 &= \mathbf{\Psi}_1^{\frac{1}{2}}(\mathbf{V}_1\mathbf{x} + \mathbf{n}_1)\\ \mathbf{y}_2 &= \mathbf{\Psi}_2^{\frac{1}{2}}(\mathbf{V}_2\mathbf{x} + \mathbf{n}_2)\end{aligned} \tag{3.66}$$

with a 1 dimensional latent variable $\mathbf{x}$, where $\mathbf{x} \sim \mathcal{N}(\mathbf{x} \mid \mathbf{0}, \mathbf{I}_1)$, and independent noise variables $\mathbf{n}_1 \sim \mathcal{N}(\mathbf{n}_1 \mid \mathbf{0}, \mathbf{I}_2)$, $\mathbf{n}_2 \sim \mathcal{N}(\mathbf{n}_2 \mid \mathbf{0}, \mathbf{I}_2)$, $\mathbf{V}_1 = [2, 0]^\top, \mathbf{V}_2 = [2, 0]^\top$, $\mathbf{\Psi}_1^{\frac{1}{2}} = \begin{pmatrix} 0.1 & 0.3 \\ 0.3 & 1 \end{pmatrix}$, $\mathbf{\Psi}_2^{\frac{1}{2}} = \begin{pmatrix} 0.2 & 0.1 \\ 0.1 & 1 \end{pmatrix}$ such that the first dimensions of the data sets are correlated with each other. After training the model using two linear covariance functions, the maximum likelihood estimates for $\mathbf{\Psi}_1^{\frac{1}{2}}$ and $\mathbf{\Psi}_2^{\frac{1}{2}}$ are given by:

$$\mathbf{\Psi}_1^{\frac{1}{2}} = \begin{pmatrix} 0.1142 & 0.3273 \\ 0.3273 & 1.0783 \end{pmatrix}, \mathbf{\Psi}_2^{\frac{1}{2}} = \begin{pmatrix} 0.3102 & 0.1213 \\ 0.1213 & 0.9784 \end{pmatrix} \tag{3.67}$$

This demonstrates that the model is able to capture correlations within the data sets. The latent coordinates $\hat{\mathbf{X}}_1$ and $\hat{\mathbf{X}}_2$ are calculated and plotted against each other in Figure 3.4. For comparison, we also find the latent coordinates for the original GPLVM (which we term GPLVM-PCA) which does not capture the correlations between the data sets (where $\mathbf{\Psi} = \mathbf{I}$): we see that the estimates $\hat{\mathbf{X}}_1$ and $\hat{\mathbf{X}}_2$ are highly correlated from our model whereas those from GPLVM-PCA appear to have little structure. Thus we can use one latent coordinate estimate as the best estimate of the other's position and use this to estimate the corresponding data coordinate.

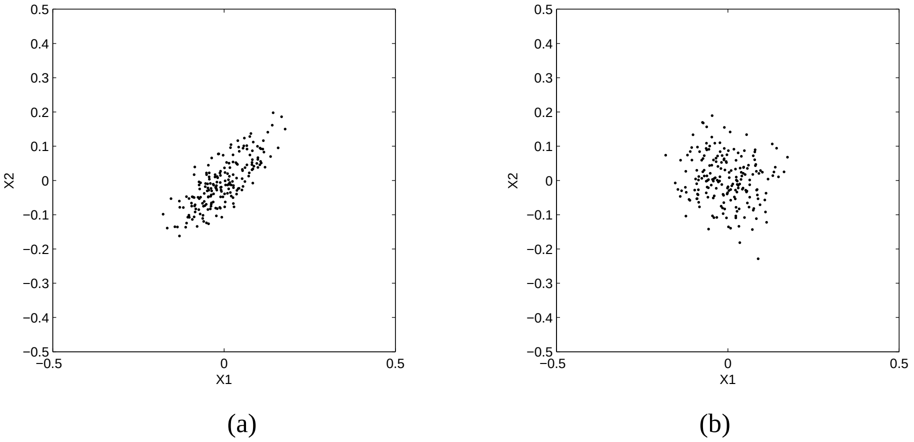


Figure 3.4: (a) The estimates of the positions of the latent points, $\hat{\mathbf{X}}_1$ and $\hat{\mathbf{X}}_2$, from each data stream using GPLVM-CCA. (b) the equivalent estimates from GPLVM-PCA.

3.6.2 Nonlinear example

To illustrate our algorithm on a nonlinear CCA problem, we create two data sets of N = 100 samples each where the nth pair of data samples is given by:

$$\mathbf{y}_{1,n} = \begin{pmatrix} 0.7 & 0.5 \\ 0.5 & 0.7 \end{pmatrix} \begin{pmatrix} \cos(0.8\mathbf{x}_n) + n_{1,1} \\ \sin(0.8\mathbf{x}_n) + n_{1,2} \end{pmatrix}$$
$$\mathbf{y}_{2,n} = \begin{pmatrix} 0.5 & -0.2 \\ -0.2 & 0.5 \end{pmatrix} \begin{pmatrix} \cos(0.8\mathbf{x}_n) + n_{2,1} \\ \sin(0.8\mathbf{x}_n) + n_{2,2} \end{pmatrix} \tag{3.68}$$

where the noise components $\mathbf{n} = [n_{1,1}, n_{1,2}, n_{2,1}, n_{2,2}]^\top \sim \mathcal{N}(\mathbf{n} \mid \mathbf{0}, \sigma_n^2\mathbf{I})$, where the noise variance $\sigma_n^2 = 0.1$, and $\mathbf{x} \in [-\pi, \pi]$. Both data sets $\mathbf{Y}_1 = [\mathbf{y}_{1,n}, ..., \mathbf{y}_{1,N}]$ and $\mathbf{Y}_2 = [\mathbf{y}_{2,n}, ..., \mathbf{y}_{2,N}]$ lie near to 1-dimensional elliptical manifolds indexed by the shared latent coordinates $\mathbf{x}$. Each data set is a linearly transformed portion of a noisy circle; there are correlations within each data set.

3.6.2.1 Training the model

We train the model (GPLVM-CCA) on the data sets using SE kernels (see (3.54)) for both GPLVM's. For comparison, we also ran the experiment for the model with the parameter $\mathbf{\Psi}$ fixed at $\mathbf{I}$ (GPLVM-PCA). This model assumes that the data in each dimension are independent of each other, and is essentially two GPLVM's (with different covariance functions) which model a data set each, and share the same set of latent coordinates. This allows us to see the effect of $\mathbf{\Psi}$ in the GPLVM-CCA model, which is learned during the optimisation. For all the experiments, we initialise the hyperparameters $\Theta_1 = \{\alpha_1, \beta_1, \gamma_1\}$ and $\Theta_2 = \{\alpha_2, \beta_2, \gamma_2\}$ of the kernels $\mathbf{K}_1$ and $\mathbf{K}_2$ respectively as $\alpha = 1, \beta = 1, \gamma = 1$, and use a 1 dimensional latent space. For the GPLVM-CCA model, we fix the scale of the kernels α_1, α_2 to 1, since the scale is already captured in $\mathbf{\Psi}_1$ and $\mathbf{\Psi}_2$. After training the models on the data, the learned hyperparameters are:

GPLVM-CCA

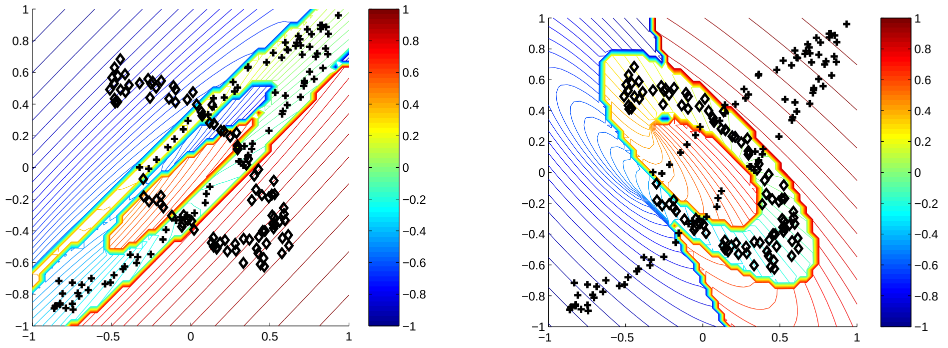
$$\begin{aligned} \alpha_1 = 1, \quad \beta_1 = 144.93, \quad \gamma_1 = 19.95, \quad \mathbf{\Psi}_1 &= \begin{pmatrix} 1.0400 & 0.9686 \\ 0.9686 & 0.9658 \end{pmatrix} \\ \alpha_2 = 1, \quad \beta_2 = 129.87, \quad \gamma_2 = 25.87, \quad \mathbf{\Psi}_2 &= \begin{pmatrix} 0.2836 & -0.2723 \\ -0.2723 & 0.3786 \end{pmatrix} \end{aligned} \tag{3.69}$$

GPLVM-PCA

$$\begin{aligned} \alpha_1 = 1.02, \quad \beta_1 = 1.77 \times 10^3, \quad \gamma_1 = 13.80, \quad \mathbf{\Psi}_1 = \mathbf{I} \\ \alpha_2 = 0.26, \quad \beta_2 = 131.59, \quad \gamma_2 = 13.59, \quad \mathbf{\Psi}_2 = \mathbf{I} \end{aligned} \tag{3.70}$$

3.6.2.2 Visualising the mapping

To visualise the mapping between latent space and data space, we plot contour maps of the estimated (1-D) latent coordinate corresponding to each data space. The lines correspond to regions of data space which project to the same latent coordinate. Denoting the latent variables underlying the two data variables $\mathbf{y}_1$ and $\mathbf{y}_2$ as $\mathbf{x}_1$ and $\mathbf{x}_2$ respectively, $\mathbf{x}_1$ is evaluated over the region $\mathbf{y}_{1,1} \in [-1, 1], \mathbf{y}_{1,2} \in [-1, 1]$ using (3.49) and $\mathbf{x}_2$ over the region $\mathbf{y}_{2,1} \in [-1, 1], \mathbf{y}_{2,2} \in [-1, 1]$ using (3.50), each on a grid of 50×50 points. The contour maps for $\mathbf{x}_1$ and $\mathbf{x}_2$ are plotted in Figures 3.5 and 3.6 for GPLVM-CCA and GPLVM-PCA respectively, along with the data sets, where $\mathbf{Y}_1$ is denoted by '+' and $\mathbf{Y}_2$ by '$\Diamond$'. For the trained GPLVM-CCA model, the projection



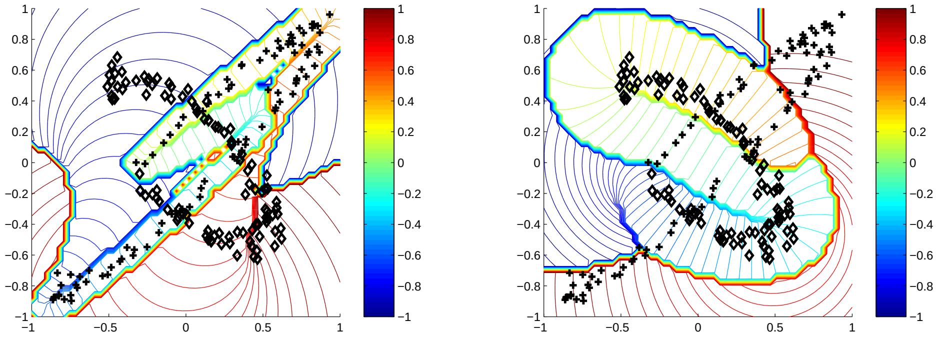
(a)

(b)

Figure 3.5: Contour plots of the latent coordinates found by GPLVM-CCA evaluated over the data space region, where $\mathbf{Y}_1$ is shown as $+$ and $\mathbf{Y}_2$ as $\diamond$. The lines correspond to regions of data space that project to the same latent coordinate. (a) shows $\mathbf{x}_1$, (b) shows $\mathbf{x}_2$

from data to latent space takes both data sets into account; Figure 3.5 illustrates how the learned manifold for each data set twists around such that the latent representation for the other set is coordinated with the first set. This is expected since we wish $\mathbf{x}_1$ and $\mathbf{x}_2$ to reflect the shared information between $\mathbf{y}_1$ and $\mathbf{y}_2$, such that $\mathbf{x}_1$ captures information about $\mathbf{y}_2$ and vice versa. This is shown particularly well in (a). In contrast, the contour maps in Figure 3.6 show that the GPLVM-PCA model does not capture shared structure with the latent coordinates.

We create a test data set $\mathbf{Y}^* = [\mathbf{Y}_1^*, \mathbf{Y}_2^*]$ by drawing a further 20 data pairs using (3.68). We evaluate the predictive power of the GPLVM-CCA and GPLVM-PCA models by predicting $\mathbf{Y}_1^*$ given $\mathbf{Y}_2^*$ and vice versa for each model. To predict $\mathbf{Y}_2^*$ given new values of the first dataset $\mathbf{Y}_1^*$, we first find $\mathbf{X}_1^*$, the set of latent coordinates underlying $\mathbf{Y}_1^*$ by solving the nonlinear optimisation problem in (3.49). Figure 3.7 shows the predictive distribution over each data set for GPLVM-PCA, (a) and (c), and GPLVM-CCA, (b) and (d), with the mean squared error per data point above each figure. The predictive distributions are found by using (3.51). As can be seen, the richer noise model of GPLVM-CCA allows the model to accurately approximate the predictive densities.



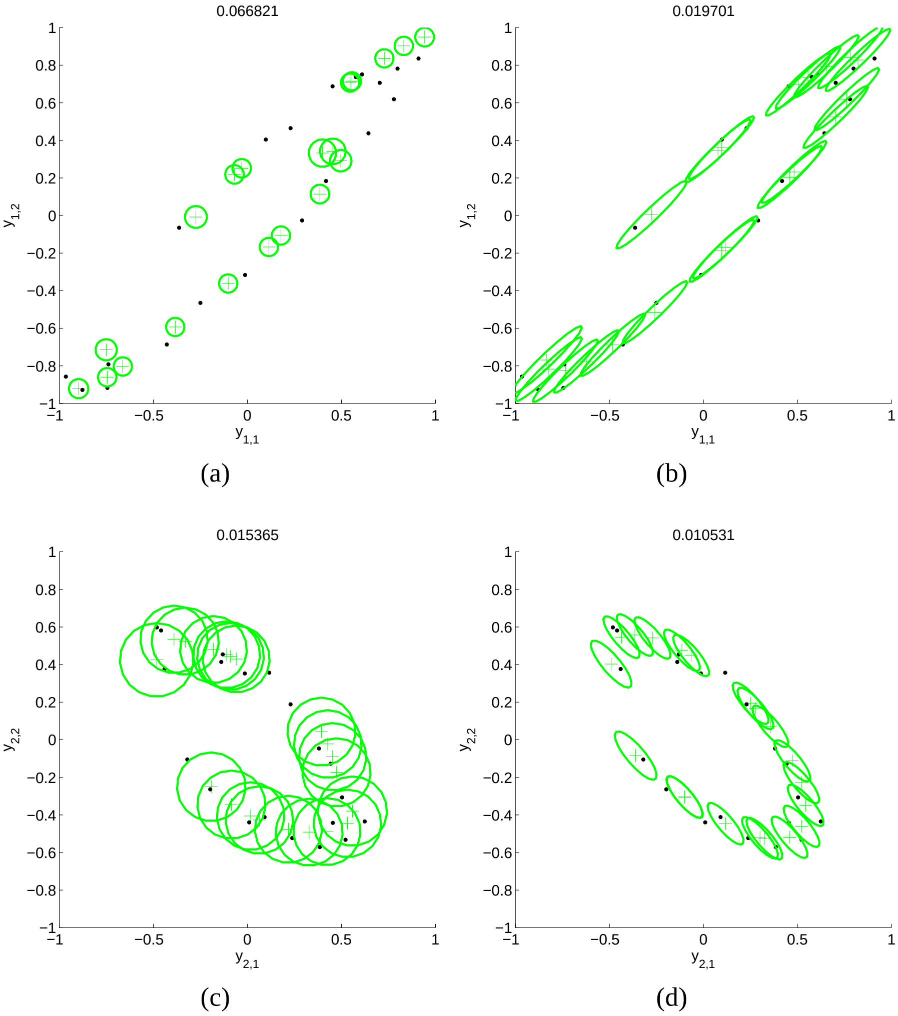
(a)

(b)

Figure 3.6: Contour plots of the latent coordinates found by GPLVM-PCA evaluated over the data space region, where $\mathbf{Y}_1$ is shown as $+$ and $\mathbf{Y}_2$ as $\diamondsuit$. The lines correspond to regions of data space that project to the same latent coordinate. (a) shows $\mathbf{x}_1$, (b) shows $\mathbf{x}_2$

3.6.3 Students' exams data

We demonstrate GPLVM-CCA's ability to find a visual representation of the shared structure between two related data sets. We test our model on a real data set from (Mardia *et al.*, 1979), which is commonly used to test the performance of CCA-type algorithms. The data set consists of 88 students' marks out of 100 on 5 exams in the subjects of Mechanics (C), Vectors (C), Algebra (O), Analysis (O), and Statistics (O), where C and O denote closed and open-book exams. We are interested in finding how highly a student's performance on closed-book exams is correlated with his performance on open-book exams. Figure 3.8 shows the set of 1-dimensional latent coordinates (each representing a student) found for both the closed-book ($\mathbf{X}_1$) and open-book exam data ($\mathbf{X}_2$). The SE kernel parameters for the trained model are $\alpha = 1.2561, \beta = 0.7773, \gamma = 1.1140$ for the closed book kernel, and $\alpha = 1.9162, \beta = 0.6940, \gamma = 1.4154$ for the open book kernel. Each student is represented by their rank number in the class, where '1' represents the student who gained the highest average score across all exams, down to '88', the lowest ranked student. There is a clear correlation between a student's performance on closed book and open book exams, which suggests that the model is able to find a representation for the students based on their ability.



(a)

(b)

(c)

(d)

Figure 3.7: Predictive densities of $\mathbf{Y}_1^*$ given $\mathbf{Y}_2^*$ (first row) and $\mathbf{Y}_2^*$ given $\mathbf{Y}_1^*$ (second row). The first column corresponds the GPLVM-PCA model, and the second column corresponds to the GPLVM-CCA model. The means of the predictive densities are shown as $+$, the data is shown as $\bullet$ and 2 st.dev of the noise covariance is plotted for each data point.

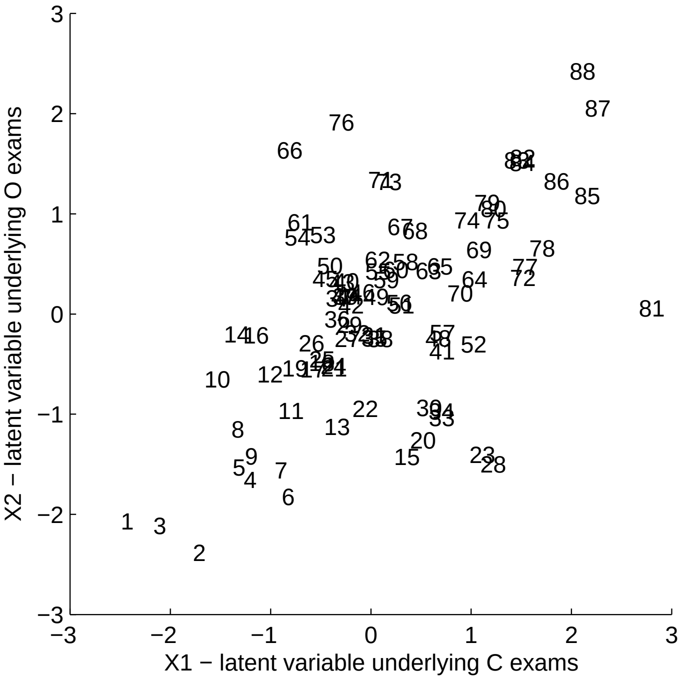


Figure 3.8: Correlated latent coordinates for closed book (C) and open book (O) exams (l) for the group of students ranked from '1' to '88'. Each latent coordinate represents a student.

Another feature of this method is that we can visually identify students who perform very well on one type of exam but less well on the other. For example we see that student 81 does very much better on the closed book exams than she does on the open book exams while the opposite is true of students 66 and 76. In this case, we can easily corroborate this fact from the original data sets but this is a useful feature for exploratory data investigation of much higher dimensional data sets.

3.6.4 Image data

We demonstrate the performance of the GPLVM-CCA model on a set of image pairs which share the same underlying degrees of freedom. We use the Frey face dataset (which can be found at `http://www.cs.toronto.edu/~roweis`) which consists of consecutive frames from a digital movie. The data set contains 1965 grayscale images of a single person's face at a resolution of 20×28. We split each image vertically in half to gain two sets of images, where $\mathbf{Y}_1$ is the set of left half images, and $\mathbf{Y}_2$ is the set of right half images. The sets of images share a complex relationship due to

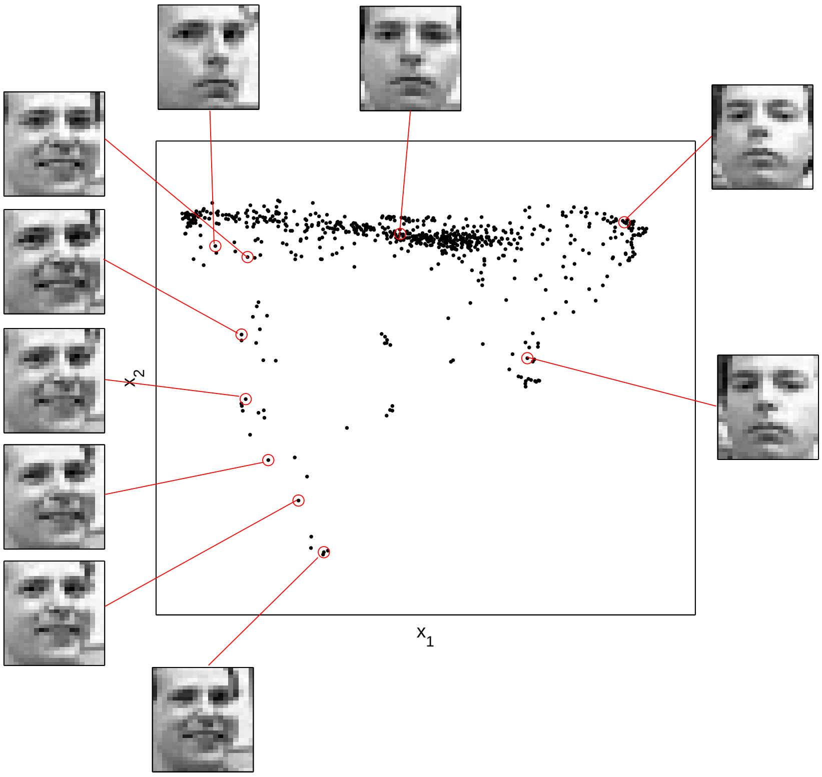


Figure 3.9: The joint structure of the pairs of face images visualised in a 2D latent space. The latent coordinate set $\mathbf{X}$ is shown with some of the corresponding image pairs.

the interaction between the left and right halves of the face to create different poses and expressions. We train the GPLVM-CCA model on a training set of 600 image pairs, using polynomial covariance functions and a 2 dimensional latent space. Due to the large size of the data set, we use a sparse approximation to the model by using the informative vector machine (IVM) (Lawrence *et al.*, 2003), which represents the model by a smaller subset of input points that contain the most information about the relationship between the two data sets. For more information about using the IVM with the GPLVM see (Lawrence, 2004, 2005). Figure 3.9 shows the shared latent coordinate set $\mathbf{X}$ underlying both data sets, shown with some corresponding image pairs, after training the model. As can be seen from the plot, the positions of the latent coordinates are determined by the pose and facial expressions. After modelling the shared underlying structure to the two sets of image pairs, we can use the trained model to make predic-

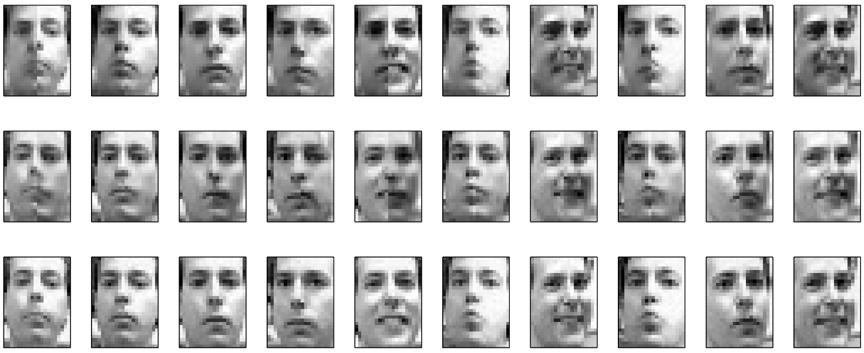


Figure 3.10: Predicted image halves given the other halves. Top row: The mean of the predictive distribution over the left half of the image given the right half. Middle row: The mean of the predictive distribution over the right half of the image given the left half. Bottom row: the true set of test images

tions about one set of images given the other. We show predictions of unknown image halves for 10 test image pairs in Figure 3.10. The first two rows show the mean of the predictive distribution over the unknown half of the 10 test images after presenting only the other image half to the model (top row: left given right, middle row: right given left). The bottom row shows the true images. As can be seen from the figure, the model is able to infer the missing image halves.

3.6.5 Image data with missing values

Since the model defines a probability density over the two sets of data variables, it is possible to handle missing values in a principled way. In this section, we show how the trained model of the previous section can be used to predict unknown image halves when presented with the other image halves that have data values missing at random. If we define an incomplete data point (for the first data set) as $\mathbf{y}_1 = \{\mathbf{y}_1^O, \mathbf{y}_1^M\}$, where $\mathbf{y}_1^O$ denotes the observed data dimensions and $\mathbf{y}_1^M$ denotes the missing data dimensions, then the likelihood for the observed data dimensions $\mathbf{y}_1^O$ given the training data and corresponding latent coordinates $\mathcal{D} = \{\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{X}\}$:

$$p(\mathbf{y}_1^O \mid \mathcal{D}, \mathbf{x}) = \int p(\mathbf{y}_1^O, \mathbf{y}_1^M \mid \mathcal{D}, \mathbf{x}) d\mathbf{y}_1^M \tag{3.71}$$

where $\mathbf{x}$ is the corresponding latent coordinate for $\mathbf{y}_1^O$. We make the approximation that

$$p(\mathbf{y}_1^O, \mathbf{y}_1^M \mid \mathcal{D}, \mathbf{x}) = p(\mathbf{y}_1^O, \mid \mathcal{D}, \mathbf{x}) p(\mathbf{y}_1^M \mid \mathcal{D}, \mathbf{x}) \quad (3.72)$$

$$= \mathcal{N}(\mathbf{y}_1^O | \mu_1^O(\mathbf{x}), \sigma_1^2(\mathbf{x}) \mathbf{\Psi}_1^O) \mathcal{N}(\mathbf{y}_1^M | \mu_1^M(\mathbf{x}), \sigma_1^2(\mathbf{x}) \mathbf{\Psi}_1^M) (3.73)$$

where

$$\mu_1^O(\mathbf{x}) = [\mathbf{k}(\mathbf{x})^\top \mathbf{K}^{-1} \mathbf{Y}_1^O]^\top \quad (3.74)$$

$$\mu_1^M(\mathbf{x}) = [\mathbf{k}(\mathbf{x})^\top \mathbf{K}^{-1} \mathbf{Y}_1^M]^\top \quad (3.75)$$

$$\sigma_1^2(\mathbf{x}) = k - \mathbf{k}(\mathbf{x})^\top \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}) \quad (3.76)$$

and $\mathbf{K} = C(\mathbf{X}, \mathbf{X}) + \beta^{-1}\mathbf{I}$, $\mathbf{k}(\mathbf{x}) = [C(\mathbf{x}, \mathbf{x}_1), ..., C(\mathbf{x}, \mathbf{x}_N)]^\top$, $k = C(\mathbf{x}, \mathbf{x})$. $\mathbf{\Psi}_1^O$ and $\mathbf{\Psi}_1^M$ are the blocks of $\mathbf{\Psi}_1$ corresponding to the observed and missing data dimensions. This ignores the correlations between the missing and data dimensions such that we can ignore the missing dimensions and find $\mathbf{x}$ to maximise

$$p(\mathbf{y}_1^O \mid \mathcal{D}, \mathbf{x}) = \mathcal{N}(\mathbf{y}_1^O | \mu_1^O(\mathbf{x}), \sigma_1^2(\mathbf{x}) \mathbf{\Psi}_1^O) \quad (3.77)$$

We can then use $\mathbf{x}$ to find the distribution over $\mathbf{y}_2$.

For this experiment, we use a test set of 100 image pairs. We remove pixel values at random from the set of left image halves, and then find the corresponding latent coordinate set, and then use this to calculate the distribution over the right image halves. We use the mean of the distribution as the predicted right half of the image; in Table 3.1 we show the root mean squared reconstruction error evaluated over the training set, for different percentages of missing data, averaged over 20 runs of each experiment. Figure 3.11 shows the predicted right image halves next to the corresponding left image halves which were presented to the algorithm (Figures 3.11a to (f)) for 10 test image pairs, when different percentages of the left image pixels were removed at random. The true images are shown in (g). As can be seen from the figure, the algorithm can

% missing pixels in left image	Average r.m.s error per pixel in right image
0	0.1299
10	0.1325
20	0.1336
30	0.1354
40	0.1372
50	0.1404
60	0.1423

Table 3.1: Pixel prediction error in the right image given the left image which has pixels missing at random. Each pixel ranges from 0 to 1.

find the corresponding right image when presented with an incomplete left image when the proportion of missing pixels is small (Figure 3.11a). As the percentage of missing pixels is increased, the model's predictive ability becomes worse (as seen in Table 3.1) which is expected. However, the test images show that the model is still able to roughly predict the underlying pose in the right image, even when up to 60% of the left image's pixels are missing at random. This is due to the learned shared embedding of the training set, as shown in Figure 3.9, in which the underlying facial pose of the image sets determines the latent coordinates' positions.

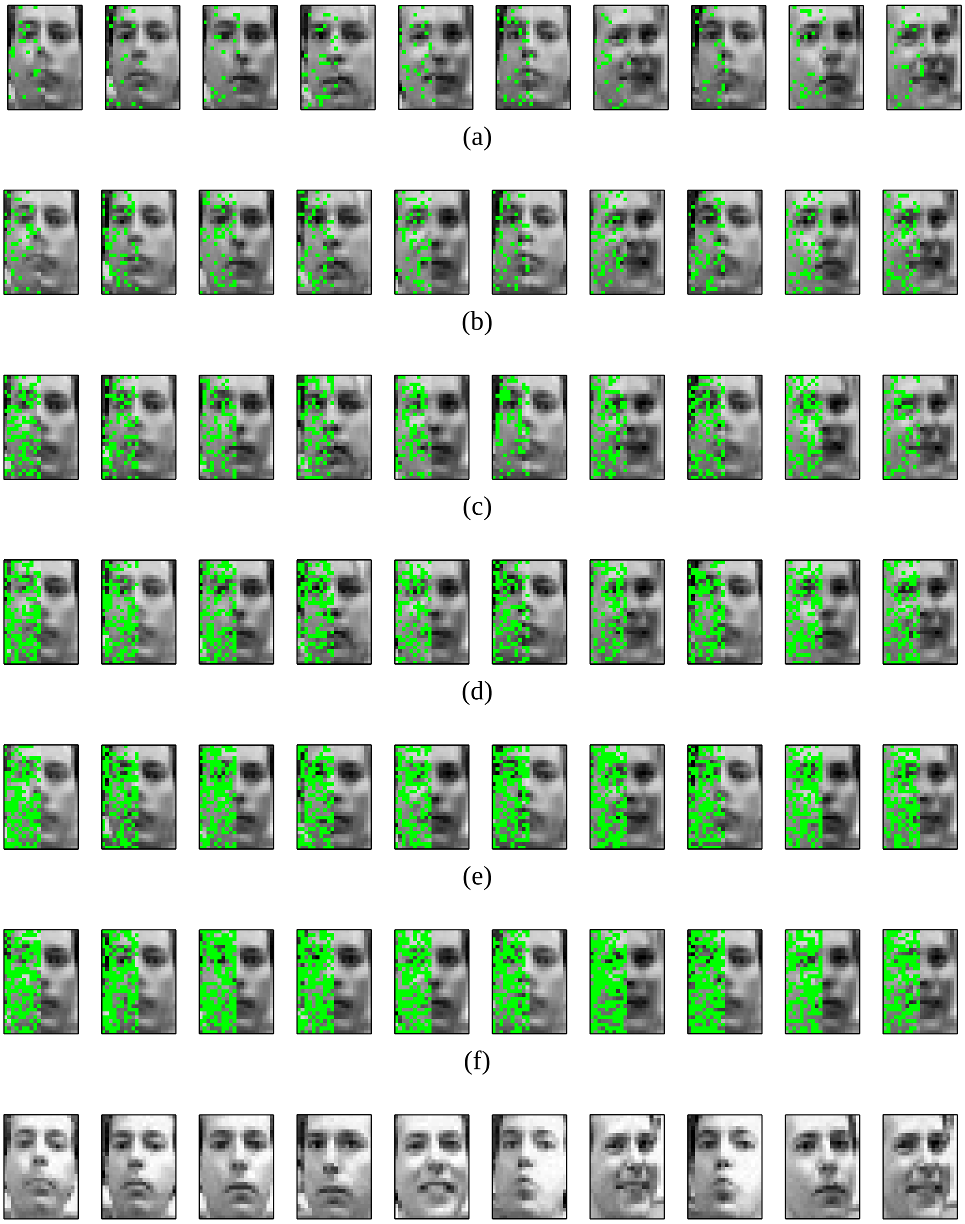
3.7 Conclusion

In this chapter, we presented a generative probabilistic framework for representing the shared structure between two related sets of data variables. Each data set is modelled as being conditionally independent of a shared set of latent variables such that the latent variable represents the features that are common to both sets of data. We also derived that the noise model for each data set has to be of sufficient flexibility to capture the within-set variation i.e. we require the noise model for the first data set $\mathbf{y}_1$ to capture the conditional entropy $H(\mathbf{y}_1 \mid \mathbf{y}_2)$ and the noise model for the second data set $\mathbf{y}_2$ to capture $H(\mathbf{y}_2 \mid \mathbf{y}_1)$. This constrains the model such that the functions underlying the data are forced to model the mutual information between the data sets.

We then showed that the within-set variation could be modelled by using linear transformations $\mathbf{\Psi}_1$ and $\mathbf{\Psi}_2$ of each data set, and showed that the generative dependency seeking model, probabilistic canonical correlation analysis (Bach & Jordan, 2005), could be interpreted within this framework. This linear model was then extended, in the spirit of the Gaussian process latent variable model (GPLVM) (Lawrence, 2004), to create a model where the shared feature space was nonlinearly related to the data spaces. Rather than having a parameterised mapping from latent to data space, Gaussian process priors were placed over the latent functions that relate the latent and data spaces. We denoted this model as GPLVM-CCA.

This model relaxes the assumption that the data dimensions within each set are independently distributed; by learning a linear transformation of underlying Gaussian processes to model each data set, the model captures the within-set variation through $\mathbf{\Psi}_1$ and $\mathbf{\Psi}_2$. This formulation can be interpreted as inducing cross covariance functions between the data dimensions within each data set (via the linear transformations $\mathbf{\Psi}_1$ and $\mathbf{\Psi}_2$) to model the private variation, such that the latent variable set is forced to model the shared relationship between the two data sets.

We then demonstrated the performance of the GPLVM-CCA model on some data sets. When using a pair of data sets that have a linear relationship, the model found sets of maximally correlated latent features for each data set. This similarity to canonical correlation analysis is due to the model's derivation from probabilistic CCA. We demonstrated the model on an example where the two data sets are nonlinearly related to a shared 1 dimensional latent space, showing that the latent space can be recovered and used to predict values of one data set given the other. Finally, we tested GPLVM-CCA's performance on a more challenging data set. We used a a set of image pairs that share the same underlying degrees of freedom. We used the Frey face dataset which consists of consecutive frames from a digital movie and split each image vertically in half to gain two sets of images. The sets of images share a complex relationship due to the interaction between the left and right halves of the face to create different poses and expressions. The GPLVM-CCA found a shared latent embedding for the two data sets that reflected the varying pose of the face throughout the data set. We then demonstrated the quality of the embedding by showing that using the trained model, one image half could be predicted when the algorithm was presented with the other image half. We also presented a mechanism for inferring the latent coordinate for an image half that contains pixels missing at random. We showed that this approximation was good enough to predict the corresponding image half with reasonable accuracy for up to 60% of missing pixels.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Figure 3.11: The predicted mean of the right half of the image given the left half of the image for 10 sets of test image pairs. The left image halves have pixels removed at random (shown in green). The % of missing left image pixels for the experiments are 10% (a), 20% (b), 30% (c), 40% (d), 50% (e), and 60% (f). The true right halves are shown in (g) along with the complete left halves.

Chapter 4

Generative models for finding shared and private structure

4.1 Introduction

In this thesis, we are interested in representing the relationship between two related data sources probabilistically. Our approach is to represent each data source as the sum of two independent components, a shared component with the other data source that captures the common information, and a private component which captures the information private to the data source. The interaction between the different components are then modelled probabilistically in terms of a generative model of the two data sources. The structure of the model reflect our assumptions about which aspects of the data are useful; in the previous chapter, we placed importance on modelling the shared features of the two data sets. After modelling the common process underlying the two data sets, the joint relationship can be compactly represented in terms of a joint latent space. This approach places less importance on modelling the components that are *private* to each data set and represents them as a noise term using a simple model.

However, there may be situations in which the shared information is not the only useful information, and interesting aspects of the data are not common to both data sets. Some useful features within one data set may not be present in the other and vice versa; this complementary property motivates the use of multiple data sources over single data sources which capture only one type of useful information. For instance, having two eyes (and two streams of visual data) allows us to gain a 3-D impression of the world. This ability of stereo vision combines both shared features and features private to each data stream to form a coherent representation of the world; common shifted features can be used in disparity estimation to infer depths of objects, while some features which may be seen in one view but not in the other, due to occlusions, can provide additional information about the scene.

If we wish to represent the private processes underlying each data set, this necessitates the use of more complex models to capture their structure. The GPLVM-CCA model that we described in Chapter 3 represents the private information for each data set with multivariate Gaussians. However, these models may not be sufficient for finding interesting features that are only present in one data set and not the other, and vice versa. In this chapter, we extend the GPLVM-CCA model and derive more complex models for the private processes. A Gaussian process latent variable model (GPLVM) prior is placed over each set of private processes, creating a flexible prior over the (optionally nonlinear) private processes. Each set of private processes is indexed by a latent space which is private to the data set, such that each ‘private’ set of latent coordinates captures the private information within its corresponding data stream.

In Section 4.2 we derive a new noise model for the GPLVM-CCA model of the previous section that is able to capture complex structure in the within-set variation. In Section 4.3 we describe how to train the model and how to infer the dimensionalities of the latent spaces, using an automatic relevance determination (ARD) procedure. In Section 4.4 we use the algorithm to perform a part-based decomposition of a synthetic image data set. The algorithm is able to represent the image data set in terms of a smaller basis of prototype images, where the basis consists of shared and private features.

4.2 Modelling a complex noise process

The GPLVM-CCA model discussed in the previous sections focuses on modelling shared information between two data sets through a shared latent space. The variation within each data set is accounted for by the mixing matrices $\mathbf{\Psi}_1^{\frac{1}{2}}$ and $\mathbf{\Psi}_2^{\frac{1}{2}}$, such that the private processes $\mathbf{n}_1$ and $\mathbf{n}_2$ are modelled as multivariate Gaussian:

$$\mathbf{n}_1 \sim \mathcal{N}(\mathbf{n}_1 \mid 0, \mathbf{\Psi}_1) \tag{4.1}$$

$$\mathbf{n}_2 \sim \mathcal{N}(\mathbf{n}_2 \mid 0, \mathbf{\Psi}_2) \tag{4.2}$$

One of the problems with these noise models is that when the dimensions m_1 and m_2 of the data sets become large, it is computationally expensive to estimate $\mathbf{\Psi}_1 \in \Re^{m_1 \times m_1}$ and $\mathbf{\Psi}_2 \in \Re^{m_2 \times m_2}$. A possible solution to this is to consider a reduced rank representation of $\mathbf{\Psi}_1$ and $\mathbf{\Psi}_2$, but this may fail to capture all of the within-set variation. A more important problem with the noise model is that it may not be sufficient for capturing complex within-set variation, since it models the private processes as noise and neglects any underlying structure.

We now review the noise model of the GPLVM-CCA model of the previous chapter, and show how it can be extended to model private processes that have underlying structure. In the GPLVM-CCA model, each dimension of the noise processes $\mathbf{n}_1$ and $\mathbf{n}_2$ can be viewed as a linear function of underlying latent variables $\mathbf{x}_1 \in \Re^{q_1}$ and $\mathbf{x}_2 \in \Re^{q_2}$ respectively such that the data is generated according to:

$$\mathbf{y}_1 = \mathbf{f}_1(\mathbf{x}) + \mathbf{n}_1 = \mathbf{f}_1(\mathbf{x}) + \mathbf{\Psi}_1^{\frac{1}{2}}\mathbf{x}_1 \tag{4.3}$$

$$\mathbf{y}_1 = \mathbf{f}_2(\mathbf{x}) + \mathbf{n}_2 = \mathbf{f}_2(\mathbf{x}) + \mathbf{\Psi}_2^{\frac{1}{2}}\mathbf{x}_2 \tag{4.4}$$

where both $\mathbf{x}_1 \in \Re^{q_1}$ and $\mathbf{x}_2 \in \Re^{q_2}$ are uncorrelated with $\mathbf{x} \in \Re^{q}$, $q_1 = m_1$, $q_2 = m_2$,

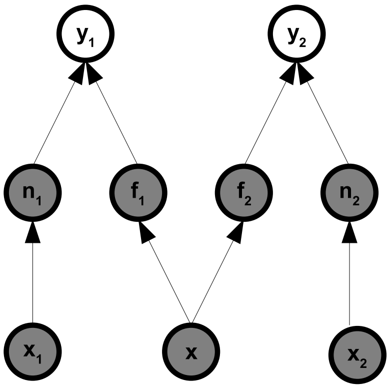


Figure 4.1: The corresponding graphical model for the unsupervised learning of two related data variables $\mathbf{y}_1$ and $\mathbf{y}_2$. Each data variable consists of two independent components, the shared function $\mathbf{f}$, and the private function $\mathbf{n}$.

and:

$$\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} \middle| \mathbf{0}, \begin{pmatrix} \mathbf{I}_{q_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{q_2} \end{pmatrix} \right) \tag{4.5}$$

Rather than restricting the noise model to linear mappings of $\mathbf{x}_1$ and $\mathbf{x}_2$, we can consider any nonlinear function, by considering noise processes of the form:

$$p(\mathbf{N}_1 \mid \mathbf{X}_1) = \prod_{i=1}^{m_1} \mathcal{N}(\mathbf{N}_1(:,i) \mid 0, \mathbf{K}_{n_1}) \tag{4.6}$$

$$p(\mathbf{N}_2 \mid \mathbf{X}_2) = \prod_{i=1}^{m_2} \mathcal{N}(\mathbf{N}_2(:,i) \mid 0, \mathbf{K}_{n_2}) \tag{4.7}$$

where m_1 and m_2 are the dimensionalities of $\mathbf{y}_1$ and $\mathbf{y}_2$ respectively, and we have placed Gaussian process priors on the ith columns of the noise $\mathbf{N}_1 = [\mathbf{n}_{1,1}, ..., \mathbf{n}_{1,N}]^\top$ and $\mathbf{N}_2 = [\mathbf{n}_{2,1}, ..., \mathbf{n}_{2,N}]^\top$ evaluated at $\mathbf{X}_1 = [\mathbf{x}_{1,1}, ..., \mathbf{x}_{1,N}]^\top$ and $\mathbf{X}_2 = [\mathbf{x}_{2,1}, ..., \mathbf{x}_{2,N}]^\top$ respectively. $\mathbf{K}_{n_1}$ and $\mathbf{K}_{n_2}$ are the covariance functions with respective inputs $\mathbf{X}_1$ and $\mathbf{X}_2$. This noise model captures the within-set variation with the columns of $\mathbf{X}_1$ and $\mathbf{X}_2$, rather than with $\mathbf{\Psi}$, as in the original GPLVM-CCA model.

We also place Gaussian process priors on the shared functions $\mathbf{F}_1 = [\mathbf{f}_{1,1}, ..., \mathbf{f}_{1,N}]^\top$

and $\mathbf{F}_2 = [\mathbf{f}_{2,1}, ..., \mathbf{f}_{2,N}]^\top$, as before:

$$p(\mathbf{F}_1 \mid \mathbf{X}) = \prod_{i=1}^{m_1} \mathcal{N}(\mathbf{F}_1(:,i) \mid 0, \mathbf{K}_{f_1}) \tag{4.8}$$

$$p(\mathbf{F}_2 \mid \mathbf{X}) = \prod_{i=1}^{m_2} \mathcal{N}(\mathbf{F}_2(:,i) \mid 0, \mathbf{K}_{f_2}) \tag{4.9}$$

where $\mathbf{K}_{f_1}$ and $\mathbf{K}_{f_2}$ are covariance functions whose input is the shared latent variable set $\mathbf{X}$. The resulting model is as follows, after integrating over the $\mathbf{F}$'s and $\mathbf{N}$'s:

$$p(\mathbf{Y}_1 \mid \mathbf{X}, \mathbf{X}_1) = \frac{1}{(2\pi)^{\frac{m_1 N}{2}} |\mathbf{K}_1|^{\frac{m_1}{2}}} \exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{K}_1^{-1}\mathbf{Y}_1\mathbf{Y}_1^\top)\right) \tag{4.10}$$

$$p(\mathbf{Y}_2 \mid \mathbf{X}, \mathbf{X}_2) = \frac{1}{(2\pi)^{\frac{m_2 N}{2}} |\mathbf{K}_2|^{\frac{m_2}{2}}} \exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{K}_2^{-1}\mathbf{Y}_2\mathbf{Y}_2^\top)\right) \tag{4.11}$$

where $\mathbf{K}_1 = \mathbf{K}_{f_1} + \mathbf{K}_{n_1}$, and $\mathbf{K}_2 = \mathbf{K}_{f_2} + \mathbf{K}_{n_2}$. Each data stream is modelled as a GPLVM, whose covariance function consists of a shared component (dependent on $\mathbf{X}$) and a private component (dependent on either $\mathbf{X}_1$ or $\mathbf{X}_2$). This is similar to the GPLVM-CCA model introduced in Chapter 3, except that the private information to each data set is now captured as a function of a private latent variable. The dimensions within each data set are modelled as independently and identically distributed, and $\mathbf{X}_1$ and $\mathbf{X}_2$ capture the correlations within $\mathbf{Y}_1$ and $\mathbf{Y}_2$ respectively. $\mathbf{X}$ captures the correlations between $\mathbf{Y}_1$ and $\mathbf{Y}_2$. The graphical representation of the model is shown in Figure 4.1.

4.2.1 Relationship to other models

The model is related to probabilistic canonical correlation analysis (PCCA) (Bach & Jordan, 2005), which can be shown by deriving the model from a GPLVM approach to PCCA. Under the PCCA model, each pair of data points $\mathbf{y} = [\mathbf{y}_1^\top, \mathbf{y}_2^\top]^\top$ is generated according to:

$$\begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \end{pmatrix} \mathbf{x} + \begin{pmatrix} \mathbf{n}_1 \\ \mathbf{n}_2 \end{pmatrix} \tag{4.12}$$

where we have assumed zero mean data. Each set of data variables is linearly related to a shared underlying latent variable $\mathbf{x} \in \Re^q$, by the matrices $\mathbf{W}_1 \in \Re^{m_1 \times q}, \mathbf{W}_2 \in \Re^{m_2 \times q}$. The noise variables $\mathbf{n}_1 \in \Re^{m_1}$ and $\mathbf{n}_2 \in \Re^{m_2}$ can be interpreted as linearly related to a set of underlying latent variables $\mathbf{x}_1 \in \Re^{m_1}$ and $\mathbf{x}_2 \in \Re^{m_2}$ respectively by matrices $\mathbf{\Psi}_1^{\frac{1}{2}} \in \Re^{m_1 \times m_1}$ and $\mathbf{\Psi}_2^{\frac{1}{2}} \in \Re^{m_2 \times m_2}$.

$$\begin{pmatrix} \mathbf{n}_1 \\ \mathbf{n}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{\Psi}_1^{\frac{1}{2}} & 0 \\ 0 & \mathbf{\Psi}_2^{\frac{1}{2}} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix}, \tag{4.13}$$

where

$$\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} \middle| \mathbf{0}, \begin{pmatrix} \mathbf{I}_{m_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{m_2} \end{pmatrix} \right) \tag{4.14}$$

We can then write that the complete set of N data points $\mathbf{Y}_1 = [\mathbf{y}_{1,1}, ..., \mathbf{y}_{1,N}]^\top$ and $\mathbf{Y}_2 = [\mathbf{y}_{2,1}, ..., \mathbf{y}_{2,N}]^\top$ are generated according to:

$$\mathbf{Y}_1 = \mathbf{X}\mathbf{W}_1^\top + \mathbf{X}_1\mathbf{\Psi}_1^{\frac{1}{2}} \tag{4.15}$$

$$\mathbf{Y}_2 = \mathbf{X}\mathbf{W}_2^\top + \mathbf{X}_2\mathbf{\Psi}_2^{\frac{1}{2}} \tag{4.16}$$

To derive the model from the PCCA model, we place conjugate Gaussian priors on the rows of $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{\Psi}_1^{\frac{1}{2}}$, and $\mathbf{\Psi}_2^{\frac{1}{2}}$:

$$p(\mathbf{W}_1) = \prod_{i=1}^{m_1} \mathcal{N}(\mathbf{w}_{1,i} \mid 0, \mathbf{I}_q) \tag{4.17}$$

$$p(\mathbf{W}_2) = \prod_{i=1}^{m_2} \mathcal{N}(\mathbf{w}_{2,i} \mid 0, \mathbf{I}_q) \tag{4.18}$$

$$p(\mathbf{\Psi}_1^{\frac{1}{2}}) = \prod_{i=1}^{m_1} \mathcal{N}(\mathbf{\Psi}_{1,i}^{\frac{1}{2}} \mid 0, \mathbf{I}_{m_1}) \tag{4.19}$$

$$p(\mathbf{\Psi}_2^{\frac{1}{2}}) = \prod_{i=1}^{m_2} \mathcal{N}(\mathbf{\Psi}_{2,i}^{\frac{1}{2}} \mid 0, \mathbf{I}_{m_2}) \tag{4.20}$$

where $\mathbf{w}_{1,i}$, $\mathbf{w}_{2,i}$, $\mathbf{\Psi}_{1,i}^{\frac{1}{2}}$ and $\mathbf{\Psi}_{2,i}^{\frac{1}{2}}$ are the ith rows of $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{\Psi}_1^{\frac{1}{2}}$, and $\mathbf{\Psi}_2^{\frac{1}{2}}$ respectively. Integrating over $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{\Psi}_1^{\frac{1}{2}}$, and $\mathbf{\Psi}_2^{\frac{1}{2}}$, we obtain the model:

$$\begin{aligned} p(\mathbf{Y}_1 \mid \mathbf{X}, \mathbf{X}_1) &= \int\int p(\mathbf{Y}_1 \mid \mathbf{X}, \mathbf{X}_1, \mathbf{W}_1, \mathbf{\Psi}_1^{\frac{1}{2}}) p(\mathbf{W}_1) p(\mathbf{\Psi}_1^{\frac{1}{2}}) d\mathbf{W}_1 d\mathbf{\Psi}_1^{\frac{1}{2}} \\ &= \frac{1}{(2\pi)^{\frac{m_1 N}{2}} |\mathbf{K}_1|^{\frac{m_1}{2}}} \exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{K}_1^{-1}\mathbf{Y}_1\mathbf{Y}_1^\top)\right) \qquad (4.21) \\ p(\mathbf{Y}_2 \mid \mathbf{X}, \mathbf{X}_2) &= \int\int p(\mathbf{Y}_2 \mid \mathbf{X}, \mathbf{X}_2, \mathbf{W}_2, \mathbf{\Psi}_2^{\frac{1}{2}}) p(\mathbf{W}_2) p(\mathbf{\Psi}_2^{\frac{1}{2}}) d\mathbf{W}_2 d\mathbf{\Psi}_2^{\frac{1}{2}} \\ &= \frac{1}{(2\pi)^{\frac{m_2 N}{2}} |\mathbf{K}_2|^{\frac{m_2}{2}}} \exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{K}_2^{-1}\mathbf{Y}_2\mathbf{Y}_2^\top)\right) \qquad (4.22) \end{aligned}$$

where $\mathbf{K}_1 = \mathbf{X}\mathbf{X}^\top + \mathbf{X}_1\mathbf{X}_1^\top$, and $\mathbf{K}_2 = \mathbf{X}\mathbf{X}^\top + \mathbf{X}_2\mathbf{X}_2^\top$. The covariance functions are linear functions of $\mathbf{X}$, $\mathbf{X}_1$, and $\mathbf{X}_2$, but we can consider any valid (nonlinear) kernel of the inputs, which imply nonlinear mappings of $\mathbf{X}$, $\mathbf{X}_1$, and $\mathbf{X}_2$ to their respective data spaces. Using a nonparametric Bayesian prior over the private functions underlying each data space is an elegant and flexible prior over underlying private structure of the data sets. Since the resulting model can be derived from probabilistic CCA, it can be viewed as a probabilistic interpretation of nonlinear canonical correlation analysis, where the underlying structure to the within-set variation is modelled explicitly.

Another model which explicitly models the within-set variation is the dependent Gaussian process model (Boyle & Frean, 2005a) which models multiple dependent outputs using Gaussian process regression. This model assumes the existence of multiple shared and private latent processes which are combined to form the outputs. The parameterisation of the covariance functions differs from our model; convolution kernels rather than covariance functions are used for the GP's.

4.3 A GPLVM-CCA model with complex noise process

4.3.1 Training the model

Learning the model, given two sets of related data $\mathbf{Y}_1$ and $\mathbf{Y}_2$, consists of finding the latent coordinates $\mathbf{X}, \mathbf{X}_1$, and $\mathbf{X}$ and the hyperparameters $\Theta_{K_{n_i}}, \Theta_{K_{f_i}}, i = 1, 2$, of the two covariance functions $\mathbf{K}_1$ and $\mathbf{K}_2$ to maximise the log likelihood function. The log likelihood is given by:

$$\mathcal{L}_{\mathbf{Y}|\mathbf{X},\mathbf{X}_1,\mathbf{X}_2} = \mathcal{L}_{\mathbf{Y}_1|\mathbf{X},\mathbf{X}_1} + \mathcal{L}_{\mathbf{Y}_2|\mathbf{X},\mathbf{X}_2} \tag{4.23}$$

since $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are conditionally independent given $\mathbf{X}$, where the likelihood functions for $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are given by:

$$\begin{aligned}
\mathcal{L}_{\mathbf{Y}_1|\mathbf{X},\mathbf{X}_1} &= \ln p(\mathbf{Y}_1 \mid \mathbf{X}, \mathbf{X}_1) \\
&= -\frac{m_1 N}{2}\ln(2\pi) - \frac{m_1}{2}\ln|\mathbf{K}_1| - \frac{1}{2}\text{tr}(\mathbf{K}_1^{-1}\mathbf{Y}_1\mathbf{Y}_1^\top) \\
\mathcal{L}_{\mathbf{Y}_2|\mathbf{X},\mathbf{X}_2} &= \ln p(\mathbf{Y}_2 \mid \mathbf{X}, \mathbf{X}_2) \\
&= -\frac{m_2 N}{2}\ln(2\pi) - \frac{m_2}{2}\ln|\mathbf{K}_2| - \frac{1}{2}\text{tr}(\mathbf{K}_2^{-1}\mathbf{Y}_2\mathbf{Y}_2^\top)
\end{aligned} \tag{4.24}$$

where $\mathbf{K}_1 = \mathbf{K}_{\mathbf{f}_1} + \mathbf{K}_{\mathbf{n}_1}$, and $\mathbf{K}_2 = \mathbf{K}_{\mathbf{f}_2} + \mathbf{K}_{\mathbf{n}_2}$, the sum of a shared and private kernel. The optimisation is similar to before; we use scaled conjugate gradients and the GPLVM toolbox. The optimisation takes place in two steps; first we jointly optimise $\mathbf{X}$ and the parameters of the shared kernels $\Theta_{K_{f_i}}, i = 1, 2$, then we jointly optimise $\mathbf{X}_1$, $\mathbf{X}_2$ and the private kernel parameters $\Theta_{K_{n_i}}, i = 1, 2$.

4.3.1.1 Optimisation of the latent points $\mathbf{X}, \mathbf{X}_1$, and $\mathbf{X}_2$

The gradients of $\mathcal{L}_{\mathbf{Y}|\mathbf{X},\mathbf{X}_1,\mathbf{X}_2}$ with respect to $\mathbf{K}_i$ is given by:

$$\frac{\partial \mathcal{L}_{\mathbf{Y}_i|\mathbf{X},\mathbf{X}_i}}{\partial \mathbf{K}_i} = -\frac{m_i}{2}\mathbf{K}_i^{-1} + \frac{1}{2}\mathbf{K}_i^{-1}\mathbf{Y}_i\mathbf{Y}_i^\top\mathbf{K}_i^{-1} \tag{4.25}$$

The gradients of $\mathcal{L}_{\mathbf{Y}|\mathbf{X},\mathbf{X}_1,\mathbf{X}_2}$ with respect to $\mathbf{X}$ can be obtained by combining (4.25) with $\frac{\partial \mathbf{K}_{f_i}}{\partial \mathbf{X}}, i = 1, 2$ using the chain rule, where $\frac{\partial \mathbf{K}_{f_i}}{\partial \mathbf{X}}$ depends on the form of the covariance function $\mathbf{K}_{f_i}$. Similarly, the gradients of $\mathcal{L}_{\mathbf{Y}|\mathbf{X},\mathbf{X}_1,\mathbf{X}_2}$ with respect to $\mathbf{X}_i$ can be obtained by combining (4.25) with $\frac{\partial \mathbf{K}_{n_i}}{\partial \mathbf{X}_i}$. As before, we seek MAP estimates for the latent coordinates by first specifying priors over $\mathbf{X}, \mathbf{X}_1$, and $\mathbf{X}_2$.

4.3.1.2 Optimisation of $\mathbf{\Theta}_{K_{f_i}}, \mathbf{\Theta}_{K_{n_i}}, i = 1, 2$

The gradients of $\mathcal{L}_{\mathbf{Y}|\mathbf{X},\mathbf{X}_1,\mathbf{X}_2}$ with respect to $\mathbf{\Theta}_{K_{n_i}}$ can be obtained by combining (4.25) with $\frac{\partial \mathbf{K}_{n_i}}{\partial \mathbf{\Theta}_{K_{n_i}}}, i = 1, 2$ using the chain rule. Similarly, the gradients of $\mathcal{L}_{\mathbf{Y}|\mathbf{X},\mathbf{X}_1,\mathbf{X}_2}$ with respect to $\mathbf{\Theta}_{K_{f_i}}$ can be obtained by combining (4.25) with $\frac{\partial \mathbf{K}_{f_i}}{\partial \mathbf{\Theta}_{K_{f_i}}}, i = 1, 2$. As before, we constrain the parameters to be positive, and seek MAP solutions.

4.3.2 Initialisation of the latent spaces

One important problem in the implementation of the original GPLVM-CCA model is the initialisation of the latent space $\mathbf{X}$, since the algorithm may become trapped in a local minimum and fail to recover the true embedded space. When extending GPLVM-CCA to explicitly model the structure of the private processes through latent spaces $\mathbf{X}_1$ and $\mathbf{X}_2$, as we describe in this chapter, the initialisation problem becomes more difficult since the degrees of freedom of the optimisation problem is increased, due to the consideration of an additional two latent spaces. Since the variation in each data set dimension is effectively shared between the shared latent set $\mathbf{X}$ and the private latent set $\mathbf{X}_1$ or $\mathbf{X}_2$, due to $\mathbf{K}_1 = \mathbf{K}_{\mathbf{f}_1}(\mathbf{X}, \mathbf{X}) + \mathbf{K}_{\mathbf{n}_1}(\mathbf{X}_1, \mathbf{X}_1)$, and $\mathbf{K}_2 = \mathbf{K}_{\mathbf{f}_2}(\mathbf{X}, \mathbf{X}) + \mathbf{K}_{\mathbf{n}_2}(\mathbf{X}_2, \mathbf{X}_2)$, the model is very sensitive to its initialisation. In our experiments we use CCA to initialise the positions of $\mathbf{X}$, since $\mathbf{X}$ represents the shared features between $\mathbf{Y}_1$ and $\mathbf{Y}_2$. To initialise the private latent spaces, we calculate the off-subspace variances for $\mathbf{Y}_1$ and $\mathbf{Y}_2$, $\mathbf{\Psi}_1$ and $\mathbf{\Psi}_2$ respectively, which are the noise covariance matrices of the probabilistic CCA method which we reviewed in Section 2.5.4. We then find $\mathbf{X}_1$ and $\mathbf{X}_2$ by projecting the corresponding data set onto the first q_1 and q_2 dominant eigenvectors of $\mathbf{\Psi}_1$ and $\mathbf{\Psi}_2$ respectively.

4.3.3 Inferring the dimensionality of the latent spaces

A problem of dimensionality reduction methods is choosing the dimensionality of the latent space q. A too low value of q can result in the model discarding some of the important information in the data as noise, and a too high q value allows the model to fit to spurious correlations in the data. In our model, we have three different latent spaces $\mathbf{X}$, $\mathbf{X}_1$ and $\mathbf{X}_2$, which capture different parts of the data - the private information in each data set and the shared information between the data sets. The dimensionalities of the latent spaces $q < \min(m_1, m_2)$, $q_1 < m_1$, and $q_2 < m_2$, enable the model to find a compact representation of the relationship between $\mathbf{Y}_1$ and $\mathbf{Y}_2$, and the underlying structure to their within-set variation. Determining the dimensionality of the latent spaces is therefore important since this will affect how the information in the data sets is shared between $\mathbf{X}$, $\mathbf{X}_1$ and $\mathbf{X}_2$.

A solution to this problem is to use automatic relevance determination (ARD) methods, as suggested in (MacKay, 1995; Neal, 1998) from the neural networks literature, which advocates the use of continuous hyperparameters to avoid the problem of a discrete model search to find the best setting of latent dimensionality. To implement ARD in the model, the dimensionality of the latent spaces is set to a maximum value $q_{\max}$. Hyperparameters are added to the covariance functions that weight each input dimension, and a hyperprior is placed on the weights to discourage large values. Irrelevant input dimensions can then be effectively removed during the training of the model i.e. the weight of the irrelevant input goes to zero, allowing the data to be best explained with as few latent dimensions as necessary. In our experiments we use an ARD polynomial covariance function for each latent process which is given by:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \alpha \left(\mathbf{x}_i^\top \mathbf{A} \mathbf{x}_j + \gamma\right)^d + \beta^{-1}\delta_{i,j} \tag{4.26}$$

with hyperparameters $\Theta_{K_{poly}} = \{\alpha, \beta, \gamma, d, \mathbf{A}\}$, and $\mathbf{A} = \text{diag}(\mathbf{a})^2$, where $\mathbf{a} = [a_1, ...a_{q_{\max}}]^\top$ is a vector of positive values. Each element a_i is the inverse of the squared correlation length scale of the process in the ith dimension. Since $\mathbf{a}$ controls the scale of each input dimension, a small scale will cause the covariance function to become almost independent of that input, deeming it 'irrelevant' to the model. We also place a zero mean Gaussian hyperprior on $\mathbf{a}$ such that small scales are favoured.

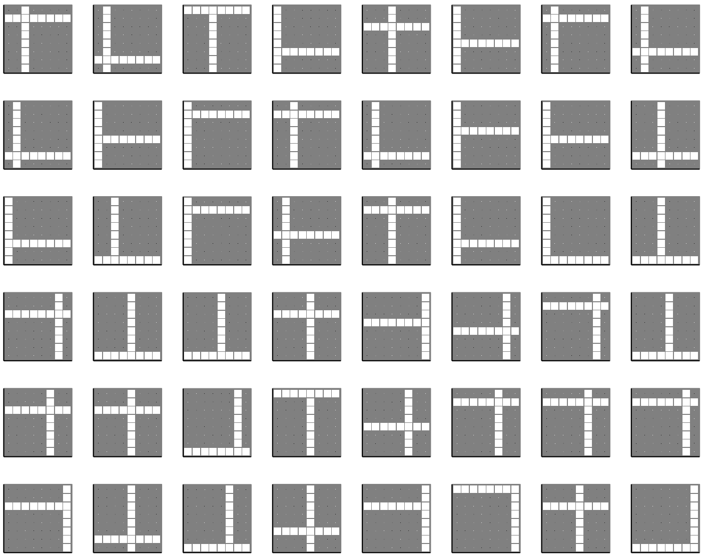


Figure 4.2: Examples of the training images. The top three rows are from the first data set (the first 24 columns of $\mathbf{Y}_1$), and the bottom three rows are from the second data set (the first 24 columns of $\mathbf{Y}_2$). Each image consists of a horizontal bar chosen at random from the 8 possibilities, which corresponds to the process shared by both sets. The first data set contains a vertical bar chosen at random from the left half of the image, and the second data set contains a vertical bar chosen from the right half of the image.

4.4 Experiments

In this section we demonstrate the model's performance on two data sets of images. We separate the images into a set of latent images. The latent images form a basis of prototype images, consisting of three sets of images, a set of images that represent the features common to both sets of data, and two sets of images that represent the features that are only present in their corresponding data set. In our experiments, we use a variation of the bars problem, which is a test problem defined in (Földiák, 1990).

4.4.1 Bars data

The bars problem e.g. (Földiák, 1990; Dayan & Zemel, 1995; Frey *et al.*, 1997; Charles & Fyfe, 1998), is a benchmark task for learning independent components from an image. While the original problem consists of decomposing a set of images into a set of underlying features (vertical and horizontal bars), in this experiment we consider a modified version of the problem that illustrates our algorithm's ability to find both shared and private features for two image sets. We create two sets of 8×8 images; 24 examples from each set are shown in Figure 4.2. Each image is generated by first instantiating one of the 8 possible horizontal bars, chosen with equal probability. For the first set of images (top three rows of Figure 4.2), one of the 4 possible vertical bars in the left half of the image is instantiated with equal probability, and similarly, for the second set of images, (bottom three rows of Figure 4.2) one of the 4 possible vertical bars in the right half of the image is instantiated with equal probability. Producing the two image sets involves a shared process in the generation of the horizontal bars, and private processes in generating the vertical bars.

Our aim is to recover the set of eight shared features - the horizontal bars - and the two sets of four private features - the vertical bars. One of the difficulties with the bars data is that each image is nonlinearly related to the underlying features (the bars), since the superposition of the features to form the image results in occlusion, or overlap, of the features. Each image can be thought of as a linear combination of horizontal and vertical bars which is then passed through a nonlinearity which models the overlap i.e. for the ith image of both data sets:

$$\mathbf{Y}_1(:,i) = G_{f_1}(\mathbf{X}\mathbf{W}_{f_1}) + G_{n_1}(\mathbf{X}_1\mathbf{W}_{n_1}) \tag{4.27}$$

$$\mathbf{Y}_2(:,i) = G_{f_2}(\mathbf{X}\mathbf{W}_{f_2}) + G_{n_2}(\mathbf{X}_2\mathbf{W}_{n_2}) \tag{4.28}$$

where G_{f_1}, G_{f_2}, G_{n_1} and G_{n_2} are nonlinear output functions, $\mathbf{W}_{f_1} \in \Re^{q \times m_1}$, $\mathbf{W}_{f_2} \in \Re^{q \times m_2}$, $\mathbf{W}_{n_1} \in \Re^{q_1 \times m_1}$ and $\mathbf{W}_{n_2} \in \Re^{q_2 \times m_2}$ are mixing matrices. For our experiment, we use polynomial covariance functions of degree 2 for each process to reflect our knowledge about the data generation process; the polynomial covariance function is given by:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \alpha \left(w\mathbf{x}_i^\top \mathbf{x}_j + \gamma \right)^2 + \beta^{-1}\delta_{i,j} \tag{4.29}$$

with hyperparameters $\Theta_{K_{poly}} = \{\alpha, \beta, \gamma, w\}$, where α is a scale parameter, β is the inverse noise variance, w controls the scale of the dot product component, and γ is a bias parameter. Polynomial kernels have proved effective for high dimensional classification problems when the input data set are binary or grayscale images i.e.(Schölkopf & Smola, 2002).

We use an 8-dimensional shared latent space $\mathbf{X}$, and a 4-dimensional private latent spaces $\mathbf{X}_1$ and $\mathbf{X}_2$ (where the columns are the underlying images). We use a training data set of 200 pairs of images (some examples are given in Figure 4.2) such that the 200 columns of $\mathbf{Y}_1 \in \Re^{64\times 200}$ and $\mathbf{Y}_2 \in \Re^{64\times 200}$ are 8×8 images that contain a vertical bar in the left and right half of the image respectively, and a horizontal bar. We also constrain the latent points' values to lie between 0 and 1, such that they correspond to underlying image pixels. Each latent point $\mathbf{x}$ is reparameterised as $\mathbf{x}'$, using a sigmoid transform $\mathbf{x} = \log(\mathbf{x}'/(1-\mathbf{x}'))$, such that the optimisation takes place in a transformed space. Figure 4.3 shows the discovered latent images (the columns of $\mathbf{X}$, $\mathbf{X}_1$, and $\mathbf{X}_2$), after training the model on the 200 pairs of training images. As can be seen, the model manages to decompose the training images into the sets of underlying shared and private features.

4.4.1.1 Reconstruction of the images

In this section, we show how the shared and private latent images which we found in the previous section can be used to reconstruct the original images. This involves finding the posterior distributions of the underlying private and shared functions given the data $\mathbf{Y}_1$ and $\mathbf{Y}_2$, and the latent features $\mathbf{X}$, $\mathbf{X}_1$ and $\mathbf{X}_2$. This investigates how well the algorithm is able to model the overlap between features. The posterior distribution over the ith column of the first set's shared underlying function (underlying the ith image of the first data set) $\mathbf{F}_1(:,i)^*$, evaluated at $\mathbf{X}^*$ given $\mathcal{D} = \{\mathbf{Y}_1(:,i), \mathbf{X}, \mathbf{X}_1\}$ is

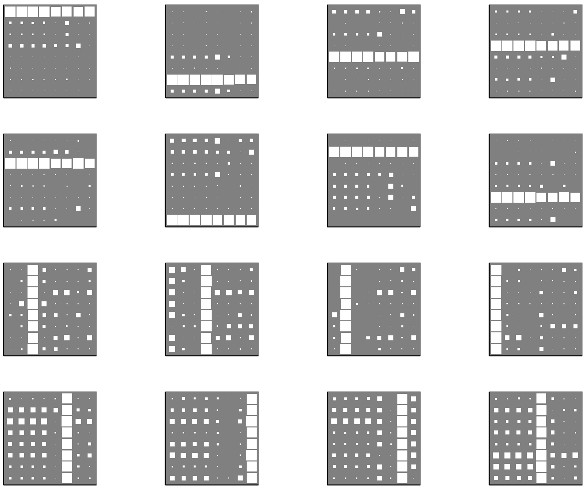


Figure 4.3: The recovered latent images. The first two rows correspond to the 8 columns of $\mathbf{X}$, and are the shared features i.e. the horizontal bars. The third row corresponds to the 4 columns of $\mathbf{X}_1$, the vertical bars in the left half of the image, and the fourth row corresponds to the 4 columns of $\mathbf{X}_2$, the vertical bars in the right half of the image.

given by:

$$p(\mathbf{F}_1(:,i)^* \mid \mathbf{X}^*, \mathcal{D}) = \mathcal{N}(\mathbf{F}_1(:,i)^* \mid \mu_{F_1}(\mathbf{X}^*), \sigma^2_{F_1}(\mathbf{X}^*)) \tag{4.30}$$

where

$$\mu_{F_1}(\mathbf{X}^*) = (\mathbf{k}_{f_1}(\mathbf{X}^*))\mathbf{K}_1^{-1}\mathbf{Y}_1(:,i) \tag{4.31}$$

$$\sigma^2_{F_1}(\mathbf{X}^*) = k_{f_1} - (\mathbf{k}_{f_1}(\mathbf{X}^*))\mathbf{K}_1^{-1}(\mathbf{k}_{f_1}(\mathbf{X}^*))^\top \tag{4.32}$$

and $\mathbf{k}_{f_1}(\mathbf{X}^*) = C_{f_1}(\mathbf{X}^*, \mathbf{X})$, where C_{f_1} denotes the first set's 'shared' kernel without the white noise variance, $\mathbf{K}_1$ is as before, and $k_{f_1} =$ diag$(C_{f_1}(\mathbf{X}^*, \mathbf{X}^*))$.

The posterior distribution over the ith column of the first set's private underlying

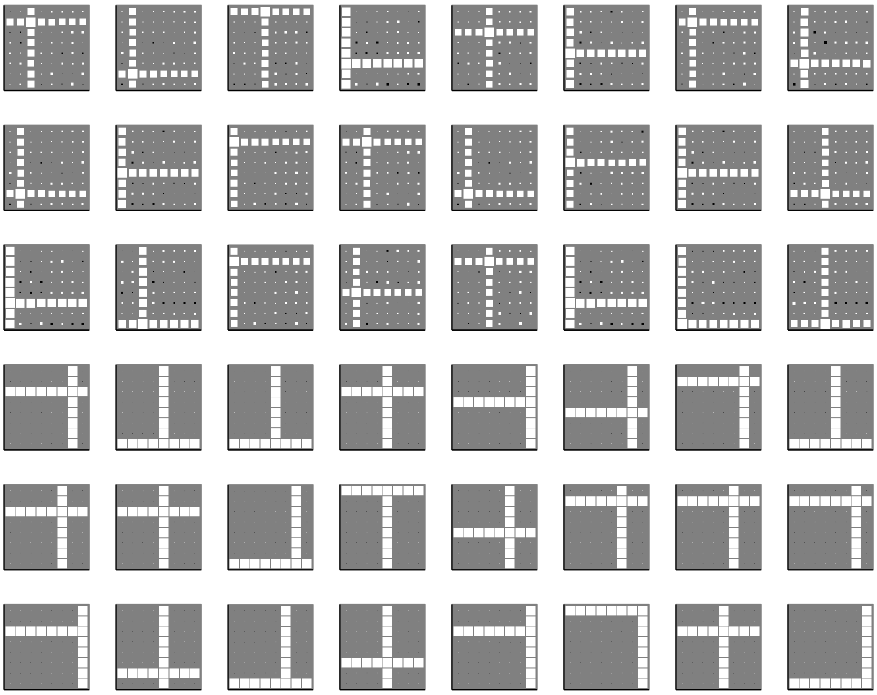


Figure 4.4: 24 reconstructed images from the first data set $\mathbf{Y}_1$ (top three rows) and the second data set $\mathbf{Y}_2$ (bottom three rows)

function $\mathbf{N}_1(:,i)^*$, evaluated at $\mathbf{X}_1^*$ given $\mathcal{D} = \{\mathbf{Y}_1(:,i), \mathbf{X}, \mathbf{X}_1\}$ is given by:

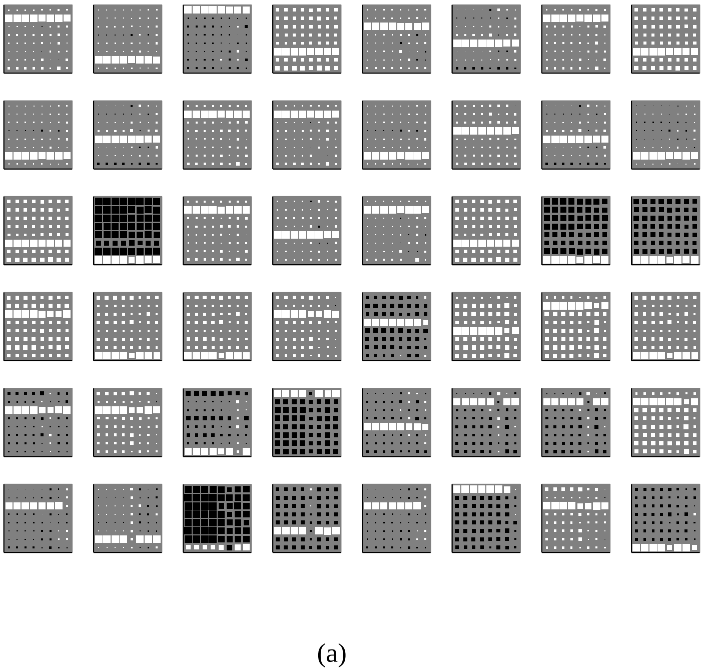
$$p(\mathbf{N}_1(:,i)^* \mid \mathbf{X}_1^*, \mathcal{D}) \quad = \quad \mathcal{N}(\mathbf{N}_1(:,i)^* \mid \mu_{N_1}(\mathbf{X}_1^*), \sigma_{N_1}^2(\mathbf{X}_1^*)) \tag{4.33}$$

where

$$\mu_{N_1}(\mathbf{X}_1^*) \quad = \quad (\mathbf{k}_{n_1}(\mathbf{X}_1^*))\mathbf{K}_1^{-1}\mathbf{Y}_1(:,i) \tag{4.34}$$

$$\sigma_{N_1}^2(\mathbf{X}^*) \quad = \quad k_{n_1} - (\mathbf{k}_{n_1}(\mathbf{X}_1^*))\mathbf{K}_1^{-1}(\mathbf{k}_{n_1}(\mathbf{X}_1^*))^\top \tag{4.35}$$

and $\mathbf{k}_{n_1}(\mathbf{X}_1^*) = C_{n_1}(\mathbf{X}_1^*, \mathbf{X}_1)$, where C_{n_1} denotes the first set's 'private' kernel without the white noise variance, and $k_{n_1} = \text{diag}(C_{n_1}(\mathbf{X}_1^*, \mathbf{X}_1^*))$. We can similarly find the posterior distributions over the shared and private functions for the second data set. We evaluate the posterior means for $\mathbf{F}_1^*$ and $\mathbf{F}_2^*$ evaluated at $\mathbf{X}$, and $\mathbf{N}_1^*$ and $\mathbf{N}_2^*$ evaluated at $\mathbf{X}_1$ and $\mathbf{X}_2$ respectively. Figure 4.4 shows the first 24 reconstructed images for each data set, given by the posterior means for $\mathbf{Y}_1^* = \mathbf{F}_1^* + \mathbf{N}_1^*$ and $\mathbf{Y}_2^* = \mathbf{F}_2^* + \mathbf{N}_2^*$



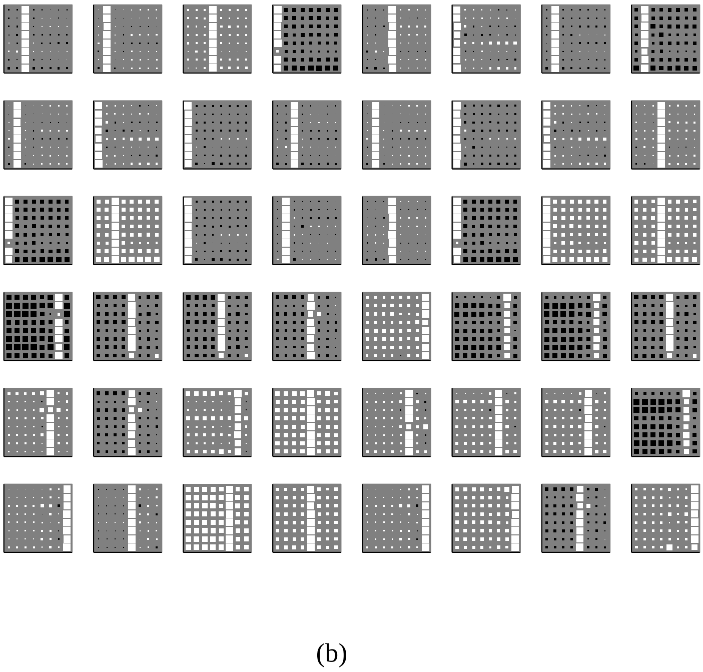


Figure 4.5: The posterior mean of the underlying shared functions is shown in (a) for the first 24 images of $\mathbf{Y}_1$ (top three rows) and $\mathbf{Y}_2$ (bottom three rows). (b) shows the posterior mean of the underlying private functions for $\mathbf{Y}_1$ (top three rows) and $\mathbf{Y}_2$ (bottom three rows)

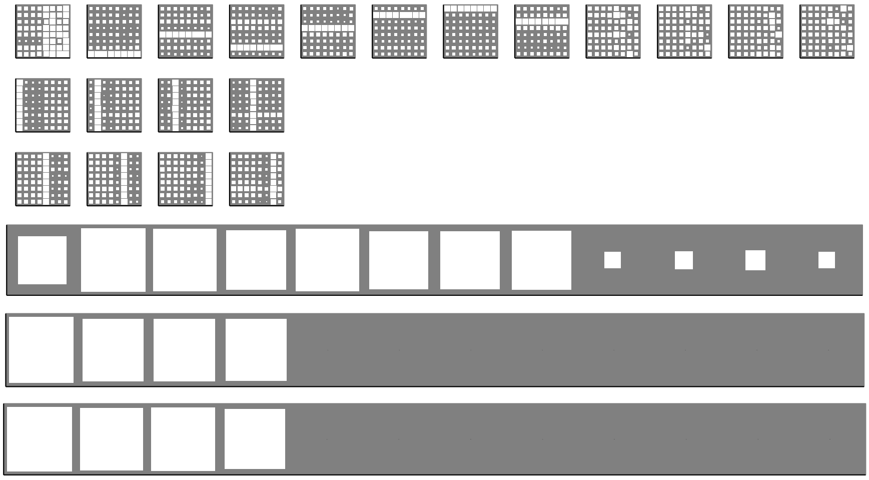


Figure 4.6: The latent images and associated scales after training the model using an ARD polynomial kernel. From top to bottom, the first three rows show the sets of latent images found for the shared space, the private space underlying $\mathbf{Y}_1$, and underlying $\mathbf{Y}_2$. The next three rows show the scales of each latent image for the shared space, and the first and second private spaces.

The top three rows are reconstructions for the first set, and the bottom three rows are reconstructions for the second set. The reconstructed images are a good approximation to the original images shown in Figure 4.2. The reconstructions for the second set model the overlap between bars more accurately than for the first set. Figure 4.5 shows the shared and private components of each image. (a) shows the posterior mean of the shared functions $\mathbf{F}_1^*$ (top three rows) and $\mathbf{F}_2^*$ (bottom three rows), and (b) shows the posterior mean of the private functions $\mathbf{N}_1^*$ (top three rows) and $\mathbf{N}_2^*$ (bottom three rows). An interesting observation is that in some of the images, a pixel is missing from one of the bars. This is due to the latent images being put through the nonlinear map implied by the polynomial covariance function. This aids in the successful reconstructions of the original image; the overlap between bars is taken into account by removing a pixel at the point in the image where the bars intersect.

4.4.2 Automatically finding the latent dimensionality of the shared space

In the previous set of experiments, we set the dimensionalities of the latent spaces - for the shared space $q = 8$, and the private spaces $q_1 = q_2 = 4$, to reflect our prior knowledge about the problem. We show how the dimensionality of the latent spaces can be automatically determined by using ARD polynomial kernels for the shared and private processes. This automatically finds the scale a_i of each input latent dimension to the kernel, such that irrelevant dimensions can be discarded ($a_i = 0$). We found that this procedure was very sensitive to the initialisation of the model because this increases the degrees of freedom of the model to the extent that the model always got stuck in a local optimum of the log likelihood function. However, we found that if we set the private spaces to the correct dimensionality $q_1 = q_2 = 4$, the model was able to correctly infer the dimensionality of the shared latent space. Figure 4.6 shows the latent images and their associated scales after setting $q = 12$, $q_1 = q_2 = 4$. As can be seen, the model correctly detects that the shared space has an intrinsic dimensionality of $q = 8$, by pruning out 4 unnecessary inputs.

4.5 Conclusion

In this chapter, we have presented a probabilistic generative framework for analysing two sets of data, where the structure of each data set is represented in terms of a shared component and a private component. In the previous chapter, we presented the GPLVM-CCA model which modelled the private (or noise) processes underlying each data set as a multivariate Gaussian. We extended this model to allow for a complex noise process that reflects the underlying structure to the within-set variation. We explicitly modelled this structure as private latent spaces for each data set, and placed Gaussian process priors over the private functions in data space. The resulting model can be interpreted as two GPLVM's, where the covariance function of each GPLVM is dependent on a shared latent space, which captures the common information, and a private latent space, which captures the private information.

We then demonstrated that the model was able to extract shared and independent components from two sets of images, which would not be possible using the GPLVM-CCA model of the previous chapter. While including a complex noise model is beneficial since it avoids an oversimplified representation of the within-set variation, the difficulty of the optimisation problem is increased because we have to optimise three latent spaces. We found that the model often became trapped in local minima during the optimisation and it was necessary to find a good initialisation of the model. We also showed that the model was able to infer the dimensionality of the shared latent space when using automatic relevance determination (ARD) kernels for the GP's.

Chapter 5

Mixture models for finding shared structure

5.1 Introduction

In this chapter, we present a model for finding a joint probabilistic representation of two data sources, which builds on work in (Fyfe & Leen, 2006). In general, existing methods for finding shared structure are discriminative methods, which find a set of features for each set that optimise a similarity measure between the features e.g. (Hotelling, 1936; Borga, 1998; Lai & Fyfe, 2000). Using these methods can be problematic; a probability density is not defined over the two sets of data variables, and therefore we cannot evaluate quantities such as the predictive density over one data set given the other. Additionally, these methods do not model the underlying data generating process. Though this may be efficient in that the modelling power is focused on optimising the quantity of interest - the similarity of the extracted features - it is difficult to incorporate prior knowledge about the feature space. With this lack of knowledge about the problem, care has to be taken in designing appropriate nonlinear mappings for finding nonlinearly related pairs of features using discriminative techniques. An inflexible mapping may not recover the true underlying shared structure between the data sets, and an overly flexible mapping may find spurious correlations between the data sets.

This problem of inferring the appropriate complexity of the model can be addressed using nonparametric Bayesian methods. The complexity of the model is allowed to grow with the number of data points such that the necessary complexity is inferred from the data. This involves placing a prior over a family of probability distributions over the data generating process to allow a flexible prior on the underlying data distribution. One such prior from the nonparametric statistics field is the Dirichlet process (DP) (Ferguson, 1973), which is a distribution over distributions. In this chapter, we assume that each data set lies close to a nonlinear manifold in data space, each indexed by a shared set of latent coordinates, which reflects the shared structure underlying the data sets. We extend the probabilistic formulation of canonical correlation analysis (PCCA) (Bach & Jordan, 2005), which we reviewed in Section 2.5.4 to a mixture of PCCA in the spirit of the mixture of probabilistic principal component analyzers (Tipping & Bishop, 1997) to find a low dimensional representation of two related data sources. The resulting model approximates the pair of nonlinear manifolds by pairs of local linear submodels. We use the DP as a nonparametric prior for the parameters of the mixture model, allowing the number of mixture components to grow with the number of data points, such that the flexibility of the manifolds is inferred from the data automatically. We call this model a Dirichlet process mixture model of probabilistic canonical correlation analysers.

In Section 5.2 we review mixture models, and derive a mixture of probabilistic canonical correlation analysers (PCCA). We show that it is not possible to infer an appropriate number of mixture components when using maximum likelihood methods. In Section 5.3 we review a Bayesian approach to the problem, and show how a finite mixture model can be generalised to an infinite mixture model by placing a nonparametric Dirichlet process on the model parameters. In Section 5.4 we present a Dirichlet process mixture model of PCCA, and evaluate the model's performance on a toy data set.

5.2 Mixtures of latent variable models

Since canonical correlation analysis (CCA) defines linear subspaces for each data space, this is insufficient for modelling the relationship between two data sets where the underlying shared structure is nonlinear. However, it may be reasonable to assume that local regions of the data spaces can be modelled by linear approximations, where the accuracy of the approximation depends on factors such as the locations of the local regions that are chosen, their size, and the strength of the nonlinearity in the data. There are a number of techniques proposed in the literature for modelling a single data set by approximating a global nonlinear structure with a combination of local principal component analysis (PCA) models. These methods are generally a two stage procedure; the data is first partitioned into local regions, and then the principal subspace is estimated within each partition. The arbitrariness in this procedure is reflected in the variety of algorithms that have been proposed i.e. (Hinton *et al.*, 1995; Bregler & Omohundro, 1995; Kambhatla & Leen, 1997), and none define a probability density. However, the probabilistic formulation of principal component analysis proposed in (Tipping & Bishop, 1999) can be naturally extended to a mixture of probabilistic principal component analyzers (Tipping & Bishop, 1997) in the probabilistic framework, overcoming the *ad hoc* nature of the previously mentioned algorithms by estimating the partitions and principal component vectors through maximisation of a single likelihood function, and defining a probability density for the model.

Following this idea, in this chapter we extend the probabilistic formulation of CCA to a mixture of PCCA in the spirit of the mixture of probabilistic principal component analyzers (Tipping & Bishop, 1997) to find a low dimensional representation of two related data sources. This models each data set as lying close to a nonlinear manifold in data space, each indexed by a shared set of latent coordinates. Corresponding local regions of each manifold are modelled by a linear approximation with a probabilistic canonical correlation analyser. Within the probabilistic framework, it is easy to extend a latent variable model to a mixture of latent variable models. A mixture model models the density for a data point $\mathbf{y}_n$ as a weighted average of K latent variable model densities, where K is the number of mixture components. The probability for $\mathbf{y}_n$ is given by:

$$p(\mathbf{y}_n \mid \theta) = \sum_{k=1}^{K} p(\mathbf{y}_n \mid \theta_k, c_n = k) p(c_n = k \mid \boldsymbol{\pi}) \tag{5.1}$$

where $c \in \{1, ..., K\}$ is a discrete variable which indicates which latent variable model has been chosen, $\boldsymbol{\pi} = [\pi_1, ..., \pi_K]^\top$ is a vector of mixing proportions (such that $\sum_{k=1}^{K} \pi_k = 1$). $p(\mathbf{c} \mid \boldsymbol{\pi})$ is a multinomial distribution over $\mathbf{c}$, where $\mathbf{c} = \{c_1, ..., c_N\}$ is the set of indicators for all N data points, such that $p(c_n = k \mid \boldsymbol{\pi}) = \pi_k$. To simplify notation, we will write $c_n = k$ as k from now on. $p(\mathbf{y}_n \mid \theta_k, k)$ is the probability of $\mathbf{y}_n$ under the kth latent variable model, with the corresponding set of parameters θ_k, and $\theta = \{\theta_1, ..., \theta_K\}$ is the complete set of parameters.

To create a mixture of probabilistic Canonical Correlation Analysers, the kth latent variable model density has the form:

$$\begin{aligned} p(\mathbf{y}_n \mid \theta_k, k) &= \int p(\mathbf{y}_n \mid \mathbf{x}_n, \theta_k, k) p(\mathbf{x}_n \mid k) d\mathbf{x}_n && (5.2) \\ &= \mathcal{N}(\mathbf{y}_n \mid \mu_k, \mathbf{W}_k \mathbf{W}_k^\top + \Psi_k) && (5.3) \end{aligned}$$

where

$$\begin{aligned} p(\mathbf{y}_n \mid \mathbf{x}_n, \theta_k, k) &= \mathcal{N}(\mathbf{y}_n \mid \mathbf{W}_k \mathbf{x}_n + \mu_k, \Psi_k) && (5.4) \\ p(\mathbf{x}_n \mid k) &= \mathcal{N}(\mathbf{x}_n \mid \mathbf{0}, \mathbf{I}_q) && (5.5) \end{aligned}$$

with $\mathbf{y}_n$ defined as the concatenation of two sets of data variables i.e. $\mathbf{y}_n = [\mathbf{y}_{1,n}^\top \mathbf{y}_{2,n}^\top]^\top$, where $\mathbf{y}_{1,n} \in \Re^{m_1}, \mathbf{y}_{2,n} \in \Re^{m_2}$ with m_1 and m_2 being the dimensions of the two data variable sets, $\mathbf{W}_k = [\mathbf{W}_{1,k}^\top \mathbf{W}_{2,k}^\top]^\top$ with $\mathbf{W}_{1,k} \in \Re^{m_1 \times q}, \mathbf{W}_{2,k} \in \Re^{m_2 \times q}$, μ_k is the bias parameter and $\mathbf{x}_n \in \Re^q$ is the corresponding shared latent variable, where q is the dimension of the latent space. The noise covariance matrix is constrained to be of block

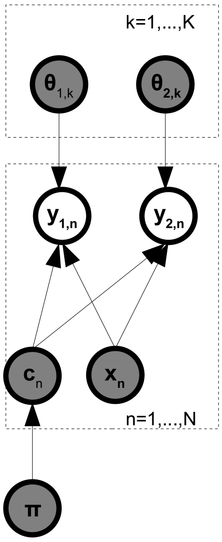


Figure 5.1: The generative model for the mixture of PCCA. A submodel k (indicated by c_n) is chosen by drawing from $p(c_n \mid \boldsymbol{\pi})$, and $\mathbf{x}_n$, the shared latent variable is drawn from $p(\mathbf{x})$. Given c_n, $\mathbf{x}_n$ and the corresponding set of parameters $\theta_{1,k} = \{\mathbf{W}_{1,k}, \mu_{1,k}, \boldsymbol{\Psi}_{1,k}\}$ and $\theta_{2,k} = \{\mathbf{W}_{2,k}, \mu_{2,k}, \boldsymbol{\Psi}_{2,k}\}$, the nth pair of data variables $\mathbf{y}_{1,n}$ and $\mathbf{y}_{2,n}$ are drawn from $p(\mathbf{y}_{1,n} \mid \mathbf{x}_n, \theta_{1,k})$ and $p(\mathbf{y}_{2,n} \mid \mathbf{x}_n, \theta_{2,k})$ respectively.

diagonal form:

$$\boldsymbol{\Psi}_k = \begin{pmatrix} \boldsymbol{\Psi}_{1,k} & 0 \\ 0 & \boldsymbol{\Psi}_{2,k} \end{pmatrix} \tag{5.6}$$

where $\boldsymbol{\Psi}_{1,k} \in \Re^{m_1 \times m_1}$, $\boldsymbol{\Psi}_{2,k} \in \Re^{m_2 \times m_2}$. We have assumed that the prior on the latent variable is the same for all K mixture components and that each Gaussian cluster has the same intrinsic dimensionality q, so from now on we will omit the indicator variable when denoting the latent priors, and rewrite $p(\mathbf{x}_n \mid k)$ as $p(\mathbf{x}_n)$. The generative model for the mixtures of probabilistic canonical correlation analyzers is shown in Figure 5.1. A pair of data points is generated by first choosing a submodel k according to $p(c_n = k \mid \boldsymbol{\pi})$, and then drawing from the kth PCCA model $p(\mathbf{y}_n \mid \theta_k, c_n = k)$.

5.2.1 EM algorithm for mixture of PCCA

The log likelihood function is given by:

$$\begin{aligned}\mathcal{L} &= \sum_{n=1}^{N} \ln p(\mathbf{y}_n \mid \theta) &(5.7)\\ &= \sum_{n=1}^{N} \ln \sum_{k=1}^{K} p(k \mid \boldsymbol{\pi}) \int p(\mathbf{y}_n \mid \mathbf{x}_n, \theta_k, k) p(\mathbf{x}_n) d\mathbf{x}_n &(5.8)\end{aligned}$$

For the kth latent variable model, the corresponding set of latent variables $\{\mathbf{x}_{k,n}\}$ is considered to be 'missing' data. As well as the latent variable sets, the indicator variables c_n, which show which submodel generated $\mathbf{y}_n$, are also 'missing'. The Expectation-Maximisation (EM) algorithm (Dempster *et al.*, 1977) can be used to handle such incomplete data problems. It finds maximum likelihood estimates of the model parameters, where the Expectation (E) step involves computing a bound on the log likelihood function by applying Jensen's inequality, followed by the Maximization (M) step, which is the standard ML calculation that would be used for a complete data model.

The expected complete data log likelihood is given by repeatedly applying Jensen's inequality to (5.8). For the kth latent variable model, if $\{\mathbf{x}_{k,n}\}$ were known, then it would be straightforward to find ML estimates of the parameters $\theta_k = \{\theta_{1,k}, \theta_{2,k}\}$. However, the joint distribution of the observed and latent variables $p(\mathbf{y}, \mathbf{x})$ is known, and the expectation of the corresponding complete data log likelihood can be calculated:

$$\mathcal{E}(\mathcal{L}_C) = \sum_{n=1}^{N} \sum_{k=1}^{K} p(\mathbf{x}_n, k \mid \mathbf{y}_n) \ln p(k \mid \boldsymbol{\pi}) p(\mathbf{y}_n \mid \mathbf{x}_n, \theta_k, k) p(\mathbf{x}_n) \qquad (5.9)$$

where $\mathcal{E}(a)$ denotes the expected value of a. The quantities $p(\mathbf{x}_n, k \mid \mathbf{y}_n) = p(\mathbf{x}_n \mid \mathbf{y}_n, k) p(k \mid \mathbf{y}_n)$ are calculated in the E step of the EM algorithm. We note that $p(\mathbf{x}_n \mid \mathbf{y}_n, k)$ is the posterior distribution over the latent variable for the kth mixture component, given the nth data point $\mathbf{y}_n$, and $p(k \mid \mathbf{y}_n) = R_{kn}$ is the posterior *responsibility* of mixture component k for generating data point $\mathbf{y}_n$, calculated as

$$R_{kn} = \frac{p(\mathbf{y}_n \mid k, \theta_k)p(k \mid \boldsymbol{\pi})}{p(\mathbf{y}_n)} = \frac{\{\int p(\mathbf{y}_n \mid \mathbf{x}, k, \theta_k)p(\mathbf{x})d\mathbf{x}\}p(k \mid \boldsymbol{\pi})}{\sum_{k=1}^{K}\{\int p(\mathbf{y}_n \mid \mathbf{x}, k, \theta_k)p(\mathbf{x})d\mathbf{x}\}p(k \mid \boldsymbol{\pi})} \tag{5.10}$$

The updates for the parameters (which aim to optimise the expected log likelihood function given in (5.9)) are as follows:

$$\tilde{\pi}_k = \frac{1}{N}\sum_{n=1}^{N} R_{kn} \tag{5.11}$$

$$\tilde{\mu}_k = \frac{\sum_{n=1}^{N} R_{kn}\mathbf{y}_n}{\sum_{n=1}^{N} R_{kn}} \tag{5.12}$$

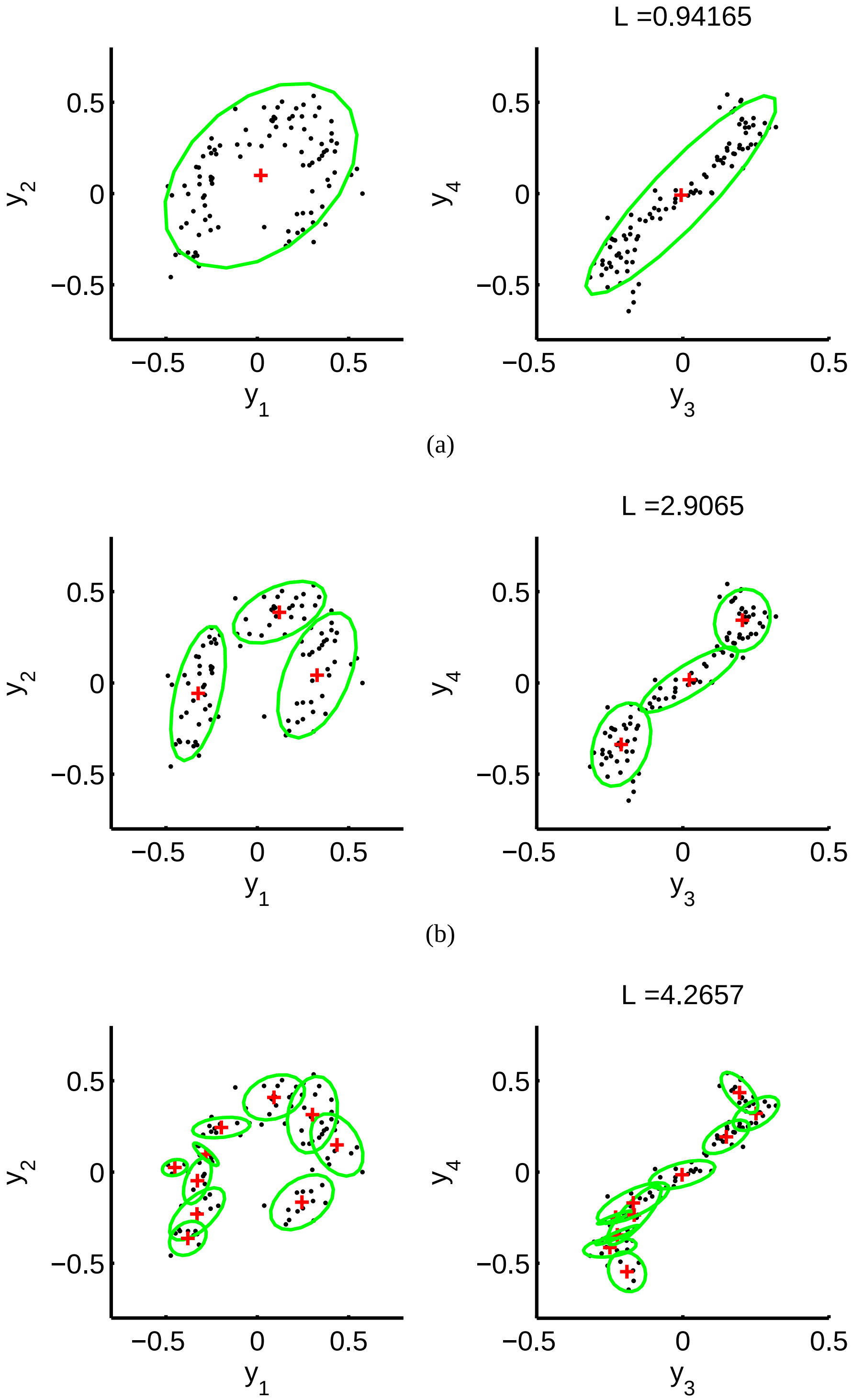
which are the standard updates for a Gaussian mixture model. For the rest of the parameter updates, we follow the approach in (Tipping & Bishop, 1997) and combine the E and M steps, gaining the intuitive result that the weights $\mathbf{W}_k$ and noise covariance $\mathbf{\Psi}_k$ can be found in terms of the local responsibility-weighted covariance matrix $\mathbf{S}_k = \frac{1}{\tilde{\pi}_k N}\sum_{n=1}^{N} R_{kn}(\mathbf{y}_n - \tilde{\mu}_k)(\mathbf{y}_n - \tilde{\mu}_k)^\top$:

$$\tilde{\mathbf{W}}_k = \mathbf{S}_k\mathbf{\Psi}_k^{-1}\mathbf{W}_k\mathbf{M}_k(\mathbf{M}_k + \mathbf{M}_k\mathbf{W}_k^\top\mathbf{\Psi}^{-1}\mathbf{S}_k\mathbf{\Psi}^{-1}\mathbf{W}_k\mathbf{M}_k)^{-1} \tag{5.13}$$

$$\tilde{\mathbf{\Psi}}_k = \begin{pmatrix} (\mathbf{S}_k - \mathbf{S}_k\mathbf{\Psi}_k^{-1}\mathbf{W}_k\mathbf{M}_k\tilde{\mathbf{W}}_k^\top)_{11} & 0 \\ 0 & (\mathbf{S}_k - \mathbf{S}_k\mathbf{\Psi}_k^{-1}\mathbf{W}_k\mathbf{M}_k\tilde{\mathbf{W}}_k^\top)_{22} \end{pmatrix} \tag{5.14}$$

where $\mathbf{M}_k = (\mathbf{I} - \mathbf{W}_k^\top\mathbf{\Psi}_k^{-1}\mathbf{W}_k)^{-1}$, and the subscripts 11 and 22 denote the upper $m_1 \times m_1$ block and the lower $m_2 \times m_2$ block on the diagonal respectively.

Figure 5.2 shows some trained mixture models of PCCA, whose parameters have been estimated by ML, using the Expectation Maximisation algorithm, where the fixed number of components are 1, 3 and 10 (see figure caption for further details). These simulations show the drawback of using maximum likelihood to evaluate the best structure for the model, since the likelihood increases with the number of components. When 10 components are used, the model can be seen to overfit the data, which is not penalised by the ML approach.



(a)

(b)

(c)

Figure 5.2: Three mixture models trained on a pair of data sets where the first data set (column 1) follows an arc, and the second data set follows a sine curve (column 2). The graphs show the plotted data (black dots) with the ML estimate of the component means (red cross) and 2 st.dev of the component noise covariance (green line). Each experiment uses a different fixed number of components, in (a) $K = 1$, which underfits the data, (b) $K = 3$, and (c) $K = 10$, which overfits the data.

5.3 Bayesian approach

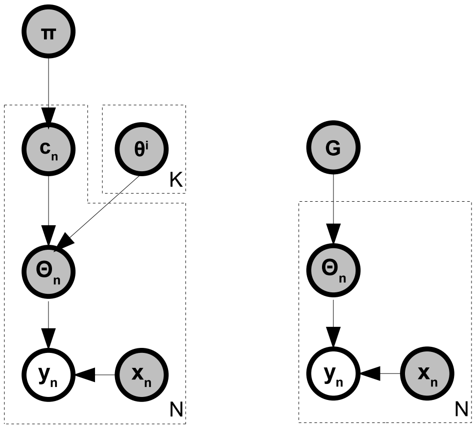
The previous section uses a maximum likelihood (ML) approach to finding the parameters of the mixture of PCCA, in which the model parameters are assigned specific values which correspond to a (local) maximum of the likelihood function. One problem with the ML framework is that there are singularities in the likelihood function, in which one or more component densities may collapse onto a single data point - the component mean becomes equal to the data point, and the corresponding covariance goes to zero - such that the model has assigned infinite density to the data point's location. This phenomenon is known as overfitting. Another problem with the maximum likelihood method is that the method does not take model complexity into account, and the data is more likely under more complex model structures, which again leads to overfitting. For instance, in the previous section it was found that the likelihood increased with the number of components in the model, such that the likelihood is maximised for the extreme case where each data point is attributed to a separate mixture component.

One approach to overcome the model selection problem unaddressed with maximum likelihood techniques is cross validation, in which a number of models, each with a separate number of components up to some maximum value, are optimised to a training set, and the predictive performance compared on an independent training set. However, this approach can be computationally expensive and does not allow for the possibility that a new data point comes from an as yet unseen component.

An elegant solution to the model selection problem is a Bayesian approach which avoids the problem of overfitting because no parameter is actually fit to the data; instead their posterior distributions are inferred, and used to make predictions for new data points. By integrating out those parameters whose cardinality scales with model complexity, more complex models are penalised since they can *a priori* model a greater range of data sets. Unfortunately when using a fully Bayesian approach, it is, in general, computationally and analytically intractable to perform the required integrals. There are several Bayesian approaches to mixture modelling in the literature which approximate the integrals required for Bayesian inference, using sampling techniques

(Neal, 1991; Rasmussen, 2000), and variational approximations (Ghahramani & Beal, 2000; Corduneanu & Bishop, 2001). In these models, the number of components is found automatically. One approach is to set a maximum number of potential components, and then when the model is trained to some data, unwanted components are suppressed, such as in (Corduneanu & Bishop, 2001), where the parameters of each Gaussian component and the latent variables are integrated out, using variational techniques, to calculate an approximation to the marginal likelihood, and the mixing coefficients are optimised using type II maximum likelihood. Similarly, in (Ghahramani & Beal, 2000), variational approximations to a full Bayesian integration over the model parameters are derived for a Bayesian mixture of factor analyzers. However, rather than starting with a maximum number of potential components, the model is initialized with a single component, and the number of components that fit the training data is found by adding new components through a stochastic procedure, and removing zero responsibility components when necessary.

Another way to address the model selection problem is to use nonparametric Bayesian techniques, in which Bayesian models with an infinite number of parameters are considered, such as the infinite mixture of Gaussians in (Rasmussen, 2000). This allows the model to be of the necessary complexity through considering a continuum of models and averaging with respect to all of these simultaneously, rather than controlling the complexity through limiting the number of parameters. Modelling data as coming from an infinite mixture has been seen to work well in the infinite mixture of Gaussians when there are only a small finite number of components in the actual mixture. The infinite mixture of Gaussians is similar to existing models in nonparametric statistics known as Dirichlet process mixture models (Ferguson, 1973; Antoniak, 1974; Escobar, 1994) but derives the model as a limiting case of a finite mixture model rather than from the Dirichlet process itself such as in (West *et al.*, 1994).



(a) (b)

Figure 5.3: Two perspectives on the finite mixture model

5.3.1 Dirichlet process mixture models

The Dirichlet process (DP) is a nonparametric distribution on distributions, or equivalently, a measure on measures (Ferguson, 1973). A DP is parameterised by a scaling parameter $\alpha_0 > 0$, and a base measure G_0. In Section 2.3.2 we reviewed the Dirichlet process and its different perspectives and showed how it could be used to place a distribution over the distribution for a parameter set Θ. We now show how to incorporate an observation model for when Θ is not observed directly, and use the DP as a nonparametric prior on the components of a mixture of probabilistic canonical correlation analyzers. This follows the approach described in (Rasmussen, 2000) and the resultant model is an infinite mixture of canonical correlation analyzers. This overcomes the model selection problems with the maximum likelihood method detailed in Section 5.2.

5.3.1.1 The finite mixture model

We interpret the parameter setting for each data point as a random variable which is drawn from a measure over the parameter space. Going back to the finite mixture model that we introduced earlier, the probability of the nth pair of data points $\mathbf{y}_n$ under the kth latent variable model can be written as:

$$p(\mathbf{y}_n \mid \theta_k) = \int p(\mathbf{y}_n \mid \Theta_n) p(\Theta_n \mid c_n = k, \theta) d\Theta_n \tag{5.15}$$

where Θ_n are the parameters associated with $\mathbf{y}_n$, $p(\mathbf{y}_n \mid \Theta_n) = \int p(\mathbf{y}_n \mid \mathbf{x}_n, \Theta_n) p(\mathbf{x}_n) d\mathbf{x}_n$ and $p(\Theta_n \mid c_n = k, \theta) = \delta(\Theta_n - \theta^k) = \delta_{\theta^k}$. c_n is a discrete variable that indexes the latent variable submodels, and θ is the set of parameter values. For all K latent variable models, the distribution over Θ_n is:

$$\begin{aligned} p(\Theta_n \mid \theta, \boldsymbol{\pi}) &= \sum_{k=1}^{K} p(\Theta_n \mid c_n = k, \theta) p(c_n = k \mid \boldsymbol{\pi}) \\ &= \sum_{k=1}^{K} \pi_k \delta_{\theta^k} \qquad (5.16) \end{aligned}$$

where $\boldsymbol{\pi} = \{\pi_1, ..., \pi_K\}$ are the mixing coefficients as before, and $p(\mathbf{c} \mid \boldsymbol{\pi})$ is a multinomial distribution. The corresponding graphical model for this mixture model representation is shown in Figure 5.3a. Since the mixture model is finite, Θ_n is equal to one of the underlying θ^k, such that the subset of $\{\Theta_n\}$ that maps to θ^k is exactly the kth cluster. We can interpret this as placing a measure over the parameter space if we define $p(\Theta_n \mid \theta, \boldsymbol{\pi})$ as G, a measure. The parameter set Θ_n for each data point is independently drawn from G, as seen in Figure 5.3b. The probability of $\mathbf{y}_n$ under all K latent variable models is given by:

$$\begin{aligned} p(\mathbf{y}_n \mid \theta, \boldsymbol{\pi}) &= \int p(\mathbf{y}_n \mid \Theta_n) \left(\sum_{k=1}^{K} p(\Theta_n \mid c_n = k, \theta) p(c_n = k \mid \boldsymbol{\pi}) \right) d\Theta_n \qquad (5.17) \\ &= \int p(\mathbf{y}_n \mid \Theta_n) \left(\sum_{k=1}^{K} \pi_k \delta_{\theta^k} \right) d\Theta_n \qquad (5.18) \end{aligned}$$

5.3.1.2 Incorporating a Dirichlet process prior

We extend the finite mixture model of the previous section to allow an infinite number of components, which allows the number of *represented* components K to be determined automatically. A Dirichlet process prior is placed on G, the random measure over the parameter space,

$$G \sim DP(G \mid G_0, \alpha_0) \qquad (5.19)$$

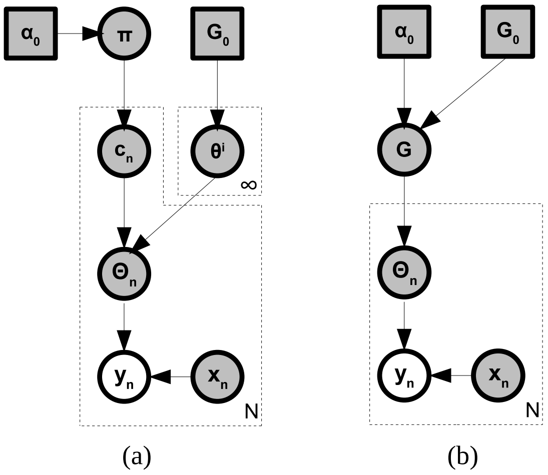


Figure 5.4: Two perspectives on the infinite mixture model

with G_0 and α_0 defined as before. The parameters for each data point $\mathbf{y}_n$ are drawn from G, as shown in Figure 5.4b (compare with Figure 5.3b). This model is a Dirichlet process mixture model. To clarify this further, we can interpret this as generalising the G of the finite case in (5.16) to the infinite case:

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta^k} \tag{5.20}$$

which is just the stick breaking representation of the distribution drawn from a Dirichlet process (Sethuraman, 1994), reviewed in Section 2.3.2.5. The parameters Θ_n for each data point take on value θ^k with probability π_k. This is equivalent to placing a prior on the mixing proportions $\boldsymbol{\pi}$ (an infinite sequence) and the parameter space θ:

$$\boldsymbol{\pi} \sim \text{Stick}(\alpha_0) \quad \theta^k \sim G_0 \tag{5.21}$$

This perspective on the infinite mixture model is visualised as a graphical model in Figure 5.4a (compare with the finite case in Figure 5.3a).

5.3.1.3 Generalising from the finite to the infinite mixture model

The Dirichlet process mixture model can be derived as the limiting case of the finite mixture model detailed in Section 5.3.1.1. Suppose that we place a symmetric Dirichlet prior on the mixing proportions of the K component mixture model $\boldsymbol{\pi} = \{\pi_1, ..., \pi_K\}$, which is conjugate to the multinomial $p(\mathbf{c} \mid \boldsymbol{\pi})$, the distribution over the indicator variables $\mathbf{c} = \{c_1, ..., c_N\}$:

$$p(\boldsymbol{\pi} \mid \alpha_0) = \text{Dir}\ (\boldsymbol{\pi} \mid \frac{\alpha_0}{K}, ..., \frac{\alpha_0}{K}) = C(\alpha_0) \prod_{k=1}^{K} \pi_k^{\alpha_0/K-1} \tag{5.22}$$

where $\alpha_0 > 0$ is a positive scaling parameter, $C(\alpha_0) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_0/K)^K}$ is a normalisation constant, and $\mathcal{E}(\pi_k) = 1/K$. Integrating out the mixing proportions we get:

$$\begin{aligned} p(c_1, ..., c_N \mid \alpha_0) &= \int p(\mathbf{c} \mid \boldsymbol{\pi}) p(\boldsymbol{\pi} \mid \alpha_0) d\boldsymbol{\pi} \\ &= \frac{\Gamma(\alpha_0)}{\Gamma(N + \alpha_0)} \prod_{k=1}^{K} \frac{\Gamma(N_k + \alpha_0/K)}{\Gamma(\alpha_0/K)} \end{aligned} \tag{5.23}$$

It is difficult to sample $\mathbf{c}$ from this distribution; instead, the indicators are Gibbs sampled to capture their dependencies. The conditional prior over the indicator variable for the nth data point given all the other indicator variables is given by:

$$p(c_n = k \mid \mathbf{c}_{-n}, \alpha_0) = \frac{N_{-n,k} + \alpha_0/K}{N - 1 + \alpha_0} \tag{5.24}$$

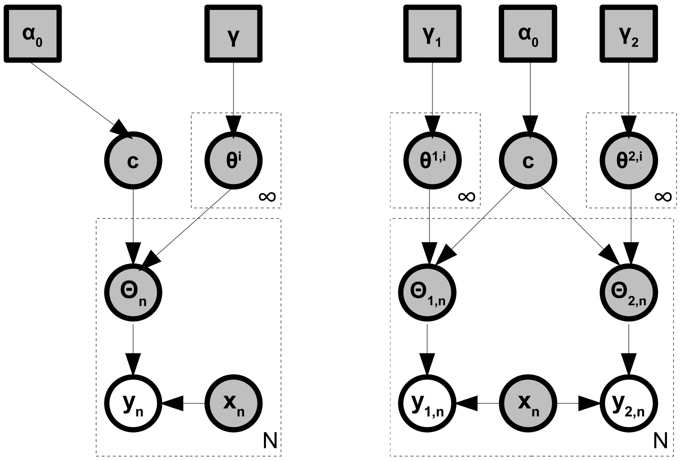
where $\mathbf{c}_{-n}$ denotes the set of indicators not including c_n, and $N_{-n,k}$ is the number of data points in the kth cluster, not including the nth data point. If we allow $K \to \infty$, i.e. we allow an infinite number of mixture components, the conditional prior on c_n becomes:

$$p(c_n = k \mid \mathbf{c}_{-n}, \alpha_0) = \frac{N_{-n,k}}{N - 1 + \alpha_0} \tag{5.25}$$

$$p(c_n \neq c_{n'} \forall n' \neq n \mid \mathbf{c}_{-n}, \alpha_0) = \frac{\alpha_0}{N - 1 + \alpha_0} \tag{5.26}$$

where the last equation is the probability that the data point is assigned to a new cluster. The parameters $\{\Theta_1, ..., \Theta_N\}$ for the data points are generated according to:

$$p(\Theta_1, ..., \Theta_N \mid \theta, \alpha_0) = \sum_{\mathbf{c}} \left(\prod_{n=1}^{N} p(\Theta_n \mid c_n, \theta) \right) p(\mathbf{c} \mid \alpha_0) \tag{5.27}$$



(a)

(b)

Figure 5.5: Graphical models for (a) the Dirichlet process mixture model and (b) the Dirichlet process mixture model of PCCA

This involves a summation over $\mathbf{c}$ i.e. over all possible assignments of data points to the components, but it is easier to evaluate in terms of the Gibbs sampling scheme as in (5.23), and if c_n takes on an existing value, then the data point n inherits the parameter set θ^{c_n}: $\Theta_n = \theta^{c_n}$. If c_n takes on a new value (starts a new cluster) then the parameter set is generated from the prior $p(\theta \mid h)$, where h is the set of hyperparameters. This is equivalent to the Pólya urn sampling scheme which we reviewed in Section 2.3.2.3. This model is a Dirichlet process mixture model, but derived in a different manner to the previous sections.

5.4 An infinite mixture of probabilistic CCA

In this section, we describe the Dirichlet process mixture model of probabilistic CCA, which uses a Dirichlet process prior on the parameters for each data point, as detailed in the previous sections.

5.4.1 Overview of the model

A DP prior is placed on the indicators $\mathbf{c} = \{c_1, ..., c_N\}$ (which show the latent submodel with which the N pairs of data points are associated), and we integrate over the mixing proportions $\boldsymbol{\pi}$. Priors are placed on the component parameters θ_k. The graphical model is shown in Figure 5.5. The probability of the data set $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_N]^\top$ is given by:

$$p(\mathbf{Y} \mid \alpha_0, \gamma) = \sum_{\mathbf{c}} \prod_{n=1}^{N} \int p(\mathbf{y}_n \mid c_n, \theta)\} p(\boldsymbol{c} \mid \alpha_0) p(\theta \mid \gamma) d\theta \tag{5.28}$$

where $p(\theta \mid \gamma)$ is the distribution over the parameter space θ (equivalent to G_0), with hyperparameters γ. This is chosen to be a conjugate prior to the probabilistic CCA likelihood. $p(\boldsymbol{c} \mid \alpha_0)$ is the distribution over the indicator variables, where the conditional priors are given in (5.25) and (5.26), the Pólya urn scheme. $p(\mathbf{y}_n \mid c_n, \theta)$ is the likelihood for a data point under the c_nth latent submodel in the probabilistic CCA model. When $c_n = k$, this is written as: $p(\mathbf{y}_n \mid c_n = k, \theta^k) = \int p(\mathbf{y}_n \mid \mathbf{x}_n, \theta^k, c_n = k) p(\mathbf{x}_n) d\mathbf{x}_n$. We can write the probability of the data set in terms on the K represented clusters:

$$\begin{aligned} p(\mathbf{Y} \mid \alpha_0, \gamma) &= \sum_{\mathbf{c}} \prod_{k=1}^{K} \left(\prod_{n:c_n=k} \int p(\mathbf{y}_n \mid \theta^k) p(\theta^k \mid \gamma) d\theta^k \right) p(\mathbf{c} \mid \alpha_0) && (5.29) \\ &= \sum_{\mathbf{c}} \prod_{k=1}^{K} \left(\int p(\mathbf{Y}^k \mid \theta^k, \mathbf{c}) p(\theta^k \mid \gamma) d\theta^k \right) p(\mathbf{c} \mid \alpha_0) && (5.30) \end{aligned}$$

where $p(\mathbf{Y}^k \mid \theta^k, \mathbf{c})$ is the probability of all the data pairs assigned to the kth cluster, given the assignments $\mathbf{c}$ of all the data, parameterised by θ^k. Additionally, we define separate parameters and hyperparameters for the two data sets $\mathbf{Y}_1$ and $\mathbf{Y}_2$ such that we can write:

$$p(\mathbf{Y} \mid \alpha_0, \gamma) = p(\mathbf{Y}_1 \mid \alpha_0, \gamma_1) p(\mathbf{Y}_2 \mid \alpha_0, \gamma_2) \tag{5.31}$$

where for $i = 1, 2$

$$p(\mathbf{Y}_i \mid \alpha_0, \gamma_i) = \sum_{\mathbf{c}} \prod_{k=1}^{K} \left(\int p(\mathbf{Y}_i^k \mid \theta^{i,k}, \mathbf{c}) p(\theta^{i,k} \mid \gamma_i) d\theta^{i,k} \right) p(\mathbf{c} \mid \alpha_0) \tag{5.32}$$

where $\mathbf{Y}_i^k$ is the kth cluster of the ith data set, $\theta^{i,k}$ is the set of parameters for the kth latent submodel for the ith data set, governed by the set of hyperparameters γ_i. The graphical model for this configuration of the parameter priors is given in Figure 5.5b, clearly showing the shared structure of a data pair $[\mathbf{y}_{1,n}, \mathbf{y}_{2,n}]$. With this formulation, it is easy to see how to compute the posterior distributions over the indicators $\mathbf{c}$, the parameters $\theta = \{\theta^1, ..., \theta^K\}$, and the hyperparameters γ and α_0.

5.4.1.1 Posterior over the parameters

The posterior distributions over the kth set of parameters are given by:

$$p(\theta^{1,k} \mid \mathbf{Y}^{1,k}, \mathbf{c}, \gamma_1) \propto p(\mathbf{Y}^{1,k} \mid \theta^{1,k}, \mathbf{c}) p(\theta^{1,k} \mid \gamma_1) \tag{5.33}$$

$$p(\theta^{2,k} \mid \mathbf{Y}^{2,k}, \mathbf{c}, \gamma_2) \propto p(\mathbf{Y}^{2,k} \mid \theta^{2,k}, \mathbf{c}) p(\theta^{2,k} \mid \gamma_2) \tag{5.34}$$

5.4.1.2 Posterior over the hyperparameters

The posterior distributions over the hyperparameters given the K sets of parameters are:

$$p(\gamma_1 \mid \theta^{1,1}, ..., \theta^{1,K}) \propto \prod_{i=1}^{K} p(\theta^{1,i} \mid \gamma_1) p(\gamma_1 \mid \xi_1) \tag{5.35}$$

$$p(\gamma_2 \mid \theta^{2,1}, ..., \theta^{2,K}) \propto \prod_{i=1}^{K} p(\theta^{2,i} \mid \gamma_2) p(\gamma_2 \mid \xi_2) \tag{5.36}$$

where $p(\gamma_1 \mid \xi_1)$ and $p(\gamma_2 \mid \xi_2)$ are vague priors over the hyperparameters, parameterised by ξ_1 and ξ_2.

5.4.1.3 Posterior over the indicators

The conditional posterior distribution over the indicators is given by:

$$p(c_n = k \mid \mathbf{c}_{-n}, \mathbf{y}_n, \theta^k) \propto p(\mathbf{y}_n \mid \theta^k, c_n = k) p(c_n = k \mid \mathbf{c}_{-n}, \alpha_0) \tag{5.37}$$

5.4.2 Graphical model

The complete graphical model for the Dirichlet process mixture model of probabilistic CCA is shown in Figure 5.6, illustrating the layered structure of the hierarchical priors. Each pair of data observations $\mathbf{y}_n = \{\mathbf{y}_{1,n}, \mathbf{y}_{2,n}\}$ is generated from one of the K rep-

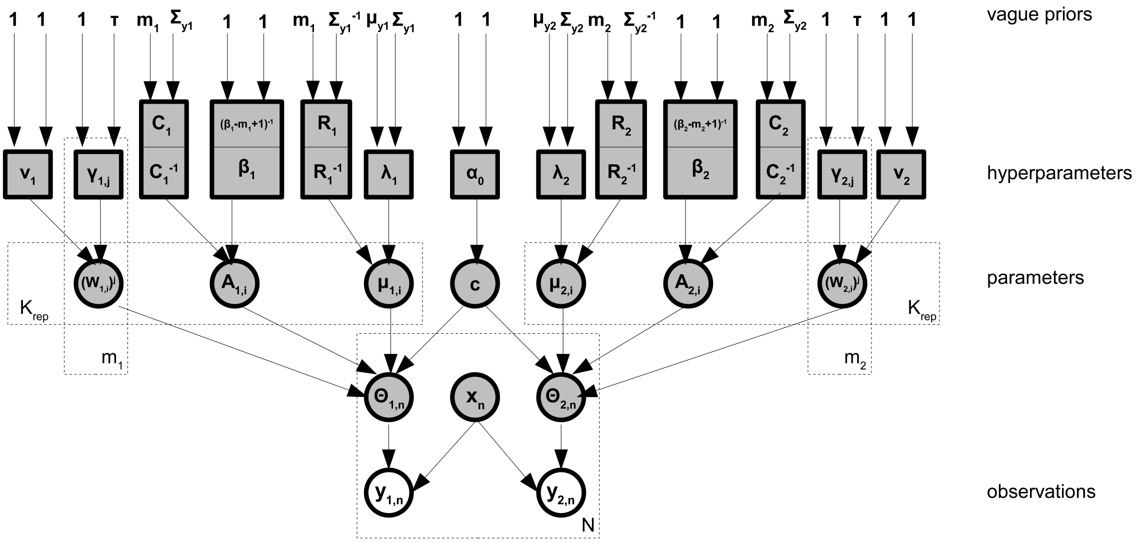


Figure 5.6: The complete graphical model for the Dirichlet process mixture model of probabilistic CCA.

resented pairs of mixture components, which is indicated by c_n. Each pair of mixture components is governed by a set of parameters, where the kth component pair's parameters are $\theta^{1,k} = \{\mu_{1,k}, \mathbf{A}_{1,k}, \mathbf{W}_{1,k}\}$ and $\theta^{2,k} = \{\mu_{2,k}, \mathbf{A}_{2,k}, \mathbf{W}_{2,k}\}$. The parameter sets are governed by a set of hyperparameters γ_1 and γ_2, which in turn are governed by vague priors ξ_1 and ξ_2. The model and a Gibbs sampling scheme is derived in the next section in detail.

5.4.3 Priors and posteriors over the component parameters and their hyperparameters

5.4.3.1 Mean vector μ_k

The mean vector for the kth latent variable model is drawn from a Gaussian distribution with hyperparameters λ and $\mathbf{R}$ which are common to all components.

$$\mu_{1,k} \sim \mathcal{N}(\mu_{1,k} \mid \lambda_1, \mathbf{R}_1^{-1}) \tag{5.38}$$

$$\mu_{2,k} \sim \mathcal{N}(\mu_{2,k} \mid \lambda_2, \mathbf{R}_2^{-1}) \tag{5.39}$$

The posterior distribution over the mean vector $\mu_{1,k}$ is given by combining the likelihood function for $\mu_{1,k}$ with the prior:

$$p(\mu_{1,k} \mid \mathbf{Y}_1, \mathbf{X}, \mathbf{c}, \lambda_1, \mathbf{R}_1) \propto p(\mathbf{Y}_1 \mid \mathbf{X}, \mathbf{c}, \theta^{1,k}) p(\mu_{1,k} \mid \lambda_1, \mathbf{R}_1) \tag{5.40}$$

We can write the likelihood $p(\mathbf{Y}_1 \mid \mathbf{X}, \mathbf{c}, \theta^{1,k})$ in terms of $\mu_{1,k}$ as:

$$\begin{aligned} p(\mathbf{Y}_1 \mid \mathbf{X}, \mathbf{c}, \theta^{1,k}) &= \prod_{n:c_n=k} \mathcal{N}(\mathbf{y}_{1,n} \mid \mathbf{W}_{1,k}\mathbf{x}_n + \boldsymbol{\mu_{1,k}}, \mathbf{\Psi}_{1,k}) && (5.41) \\ &\propto \prod_{n:c_n=k} \mathcal{N}(\mu_{1,k} \mid \mathbf{y}_{1,n} - \mathbf{W}_{1,k}\mathbf{x}_n, \mathbf{\Psi}_{1,k}) && (5.42) \\ &\propto \mathcal{N}\left(\mu_{1,k} \mid \bar{\mathbf{y}}_{1,k} - \mathbf{W}_{1,k}\bar{\mathbf{x}}_k, N_k^{-1}\mathbf{\Psi}_{1,k}\right) && (5.43) \end{aligned}$$

where $\bar{\mathbf{y}}_{1,k} = \frac{1}{N_k}\sum_{n:c_n=k} \mathbf{y}_{1,n}$, $\bar{\mathbf{x}}_k = \frac{1}{N_k}\sum_{n:c_n=k} \mathbf{x}_n$, and where N_k is the number of data points in the kth cluster. By combining this with the prior from (5.38) and using (5.40), the posterior distribution over $\mu_{1,k}$ is given by:

$$\begin{aligned} \mu_{1,k} \mid \theta_k, \mathbf{Y}_1 &\sim \mathcal{N}\left(\mu_{1,k} \mid \mu_{\mu_{1,k}}, \mathbf{\Sigma}_{\mu_{1,k}}\right) && (5.44) \\ \text{where } \mathbf{\Sigma}_{\mu_{1,k}} &= (N_k \mathbf{\Psi}_{1,k}^{-1} + \mathbf{R}_1)^{-1} && (5.45) \\ \mu_{\mu_{1,k}} &= \mathbf{\Sigma}_{\mu_{1,k}}(\mathbf{\Psi}_{1,k}^{-1} N_k(\bar{\mathbf{y}}_{1,k} - \mathbf{W}_{1,k}\bar{\mathbf{x}_k}) + \mathbf{R}_1\lambda_1) && (5.46) \end{aligned}$$

and similarly,

$$\begin{aligned} \mu_{2,k} \mid \theta_k, \mathbf{Y}_2 &\sim \mathcal{N}\left(\mu_{2,k} \mid \mu_{\mu_{2,k}}, \mathbf{\Sigma}_{\mu_{2,k}}\right) && (5.47) \\ \text{where } \mathbf{\Sigma}_{\mu_{2,k}} &= (N_k \mathbf{\Psi}_{2,k}^{-1} + \mathbf{R}_2)^{-1} && (5.48) \\ \mu_{\mu_{2,k}} &= \mathbf{\Sigma}_{\mu_{2,k}}(\mathbf{\Psi}_{2,k}^{-1} N_k(\bar{\mathbf{y}}_{2,k} - \mathbf{W}_{2,k}\bar{\mathbf{x}_k}) + \mathbf{R}_2\lambda_2) && (5.49) \end{aligned}$$

where $\bar{\mathbf{y}}_{2,k} = \frac{1}{N_k} \sum_{i:c_i=k} \mathbf{y}_{2,i}$. The hyperparameters λ_1, λ_2 and $\mathbf{R}_1, \mathbf{R}_2$ are given vague Normal and Wishart conjugate priors respectively,

$$\lambda_1 \sim \mathcal{N}(\lambda_1 \mid \mu_{y_1}, \mathbf{\Sigma}_{y_1}),\ \mathbf{R}_1 \sim \mathcal{W}(\mathbf{R}_1 \mid m_1, \mathbf{\Sigma}_{y_1}^{-1}) \tag{5.50}$$

$$\lambda_2 \sim \mathcal{N}(\lambda_2 \mid \mu_{y_2}, \mathbf{\Sigma}_{y_2}),\ \mathbf{R}_2 \sim \mathcal{W}(\mathbf{R}_2 \mid m_2, \mathbf{\Sigma}_{y_2}^{-1}) \tag{5.51}$$

where μ_{y_1} and $\mathbf{\Sigma}_{y_1}$ are the sample mean and covariance of the first data set $\mathbf{Y}_1$, and μ_{y_2} and $\mathbf{\Sigma}_{y_2}$ are the sample mean and covariance of the second data set $\mathbf{Y}_2$.

The posterior distribution over λ_1 given the mean vectors for all K components for the first data set is given by:

$$\begin{aligned} p(\lambda_1 \mid \mu_{1,1}, ..., \mu_{1,K}, \mathbf{R}_1) &\propto \prod_{1=1}^{K} p(\mu_{1,i} \mid \lambda_1, \mathbf{R}_1) p(\lambda_1) && (5.52) \\ &\propto \prod_{i=1}^{K} \mathcal{N}(\mu_{1,i} \mid \lambda_1, \mathbf{R}_1^{-1}) \mathcal{N}(\lambda_1 \mid \mu_{y_1}, \mathbf{\Sigma}_{y_1}) \\ &\propto \mathcal{N}(\lambda_1 \mid \frac{1}{K} \sum_{i=1}^{K} \mu_{1,i}, \frac{1}{K} \mathbf{R}_1^{-1}) \mathcal{N}(\lambda_1 \mid \mu_{y_1}, \mathbf{\Sigma}_{y_1}) \\ &\propto \mathcal{N}\left(\lambda_1 \mid \frac{\mathbf{R}_1 \sum_{i=1}^{K} \mu_{1,i} + \mathbf{\Sigma}_{y_1}^{-1} \mu_{y_1}}{K\mathbf{R}_1 + \mathbf{\Sigma}_{y_1}^{-1}}, \frac{1}{K\mathbf{R}_1 + \mathbf{\Sigma}_{y_1}^{-1}}\right) \\ & && (5.53) \end{aligned}$$

Similarly, the posterior distribution over λ_2 is given by:

$$\lambda_2 \mid \mu_{2,1}, ..., \mu_{2,K}, \mathbf{R}_2 \sim \mathcal{N}\left(\lambda_2 \mid \frac{\mathbf{R}_2 \sum_{i=1}^{K} \mu_{2,i} + \mathbf{\Sigma}_{y_2}^{-1} \mu_{y_2}}{K\mathbf{R}_2 + \mathbf{\Sigma}_{y_2}^{-1}}, \frac{1}{K\mathbf{R}_2 + \mathbf{\Sigma}_{y_2}^{-1}}\right) \tag{5.54}$$

The posterior distribution over $\mathbf{R}_1$ given the mean vectors $\mu_{1,1}, ..., \mu_{1,K}$ is given by:

$$\begin{aligned} p(\mathbf{R}_1 \mid \mu_{1,1}, ..., \mu_{1,K}, \lambda_1) &\propto \prod_{1=1}^{K} p(\mu_{1,i} \mid \lambda_1, \mathbf{R}_1) p(\mathbf{R}_1) && (5.55) \\ &\propto \prod_{k} \mathcal{N}(\mu_{1,k} \mid \lambda, \mathbf{R}_1^{-1}) \mathcal{W}(\mathbf{R}_1 \mid m_1, \mathbf{\Sigma}_{y_1}^{-1}) && (5.56) \end{aligned}$$

We can write the likelihood for $\mathbf{R}_1$ in terms of a Wishart distribution:

$$\prod_k \mathcal{N}(\mu_{1,k} \mid \lambda, \mathbf{R}_1^{-1}) \propto \prod_k \mathcal{W}\left(\mathbf{R}_1 \mid D+2, \frac{D+2}{(\mu_{1,k}-\lambda_1)(\mu_{1,k}-\lambda_1)^\top}\right) \tag{5.57}$$

$$\propto \mathcal{W}\left(\mathbf{R}_1 \mid K+D+1, \frac{K+D+1}{\sum_k(\mu_{1,k}-\lambda_1)(\mu_{1,k}-\lambda_1)^\top}\right) \tag{5.58}$$

The posterior distribution over $\mathbf{R}_1$ is thus given by:

$$\begin{aligned} p(\mathbf{R}_1 \mid \mu_{1,1},...,\mu_{1,K},\lambda_1) &\propto \mathcal{W}\left(\mathbf{R}_1 \mid K+D+1, \frac{K+D+1}{\sum_k(\mu_{1,k}-\lambda_1)(\mu_{1,k}-\lambda_1)^\top}\right) \\ &\times \mathcal{W}(\mathbf{R}_1 \mid m_1, \mathbf{\Sigma}_{y_1}^{-1}) \\ &\propto \mathcal{W}(\mathbf{R}_1 \mid m_1+K, \frac{m_1+K}{\mathbf{S}_{\mu_1}+m_1\mathbf{\Sigma}_{y_1}}) \end{aligned} \tag{5.59}$$

where $\mathbf{S}_{\mu_1} = \sum_{k=1}^{K}(\mu_{1,k}-\lambda_1)(\mu_{1,k}-\lambda_1)^\top$. Similarly,

$$\mathbf{R}_2 \mid \mu_{2,1},...,\mu_{2,K},\lambda_2 \sim \mathcal{W}\left(\mathbf{R}_2 \mid m_2+K, \frac{m_2+K}{\mathbf{S}_{\mu_2}+m_2\mathbf{\Sigma}_{y_2}}\right) \tag{5.60}$$

where $\mathbf{S}_{\mu_2} = \sum_{k=1}^{K}(\mu_{2,k}-\lambda_2)(\mu_{2,k}-\lambda_2)^\top$.

5.4.3.2 Covariance matrix $\mathbf{\Psi}_{1,k}$, $\mathbf{\Psi}_{2,k}$

We work with the inverse of $\mathbf{\Psi}_{1,k}$ and $\mathbf{\Psi}_{2,k}$: $\mathbf{A}_{1,k} = \mathbf{\Psi}_{1,k}^{-1}$ and $\mathbf{A}_{2,k} = \mathbf{\Psi}_{2,k}^{-1}$. The priors over $\mathbf{A}_{1,k}$ and $\mathbf{A}_{2,k}$ are Wishart distributions:

$$\mathbf{A}_{1,k} \sim \mathcal{W}(\mathbf{A}_{1,k} \mid \beta_1, \mathbf{C}_1^{-1}) \tag{5.61}$$

$$\mathbf{A}_{2,k} \sim \mathcal{W}(\mathbf{A}_{2,k} \mid \beta_2, \mathbf{C}_2^{-1}) \tag{5.62}$$

The posterior distribution over the precision matrix $\mathbf{A}_{1,k}$ is given by combining the likelihood function for $\mathbf{A}_{1,k}$ with its prior:

$$p(\mathbf{A}_{1,k} \mid \mathbf{Y}_1, \mathbf{X}, \mathbf{c}, \beta_1, \mathbf{C}_1) \propto p(\mathbf{Y}_1 \mid \mathbf{X}, \mathbf{c}, \theta^{1,k}) p(\mathbf{A}_{1,k} \mid \beta_1, \mathbf{C}_1) \tag{5.63}$$

We can write the likelihood function in terms of a Wishart distribution over $\mathbf{A}_{1,k}$:

$$
\begin{aligned}
p(\mathbf{Y}_1 \mid \mathbf{X}, \mathbf{c}, \theta^{1,k}) &= \prod_{n:c_n=k} \mathcal{N}(\mathbf{y}_{1,n} \mid \mathbf{W}_{1,k}\mathbf{x}_n + \boldsymbol{\mu}_{\mathbf{1},\boldsymbol{k}}, \mathbf{A}_{1,k}^{-1}) && (5.64)\\
&\propto \mathcal{W}\left(\mathbf{A}_{1,k} \mid N_k + m_1 + 1, \frac{N_k + m_1 + 1}{\mathbf{S}_{y_{1,k}}}\right) && (5.65)
\end{aligned}
$$

$\mathbf{S}_{y_{1,k}} = \sum_{n:c_n=k}(\mathbf{y}_{1,n} - (\mathbf{W}_{1,k}\mathbf{x}_n + \boldsymbol{\mu}_{\mathbf{1},\boldsymbol{k}}))(\mathbf{y}_{1,n} - (\mathbf{W}_{1,k}\mathbf{x}_n + \boldsymbol{\mu}_{\mathbf{1},\boldsymbol{k}}))^\top$. Substituting this expression into (5.63), along with the prior given in (5.61), the posterior over $\mathbf{A}_{1,k}$ becomes:

$$
\begin{aligned}
p(\mathbf{A}_{1,k} \mid \mathbf{Y}_1, \mathbf{X}, \mathbf{c}, \beta_1, \mathbf{C}_1) &\propto \mathcal{W}\left(\mathbf{A}_{1,k} \mid N_k + m_1 + 1, \frac{N_k + m_1 + 1}{\mathbf{S}_{y_{1,k}}}\right)\\
&\times \mathcal{W}(\mathbf{A}_{1,k} \mid \beta_1, \mathbf{C}_1^{-1})\\
&\propto \mathcal{W}\left(\mathbf{A}_{1,k} \mid N_k + \beta_1, \frac{N_k + \beta_1}{N_k\mathbf{S}_{y_{1,k}} + \beta_1\mathbf{C}_1}\right) && (5.66)
\end{aligned}
$$

Similarly, the posterior over $\mathbf{A}_{2,k}$ is given by:

$$
p(\mathbf{A}_{2,k} \mid \mathbf{Y}_2, \mathbf{X}, \mathbf{c}, \beta_2, \mathbf{C}_2) \propto \mathcal{W}\left(\mathbf{A}_{2,k} \mid N_k + \beta_2, \frac{N_k + \beta_2}{N_k\mathbf{S}_{y_{2,k}} + \beta_2\mathbf{C}_2}\right) \quad (5.67)
$$

where $\mathbf{S}_{y_{2,k}} = \sum_{n:c_n=k}(\mathbf{y}_{2,n} - (\mathbf{W}_{2,k}\mathbf{x}_n + \boldsymbol{\mu}_{\mathbf{2},\boldsymbol{k}}))(\mathbf{y}_{2,n} - (\mathbf{W}_{2,k}\mathbf{x}_n + \boldsymbol{\mu}_{\mathbf{2},\boldsymbol{k}}))^\top$.

The hyperparameters $\beta_1, \beta_2, \mathbf{C}_1$ and $\mathbf{C}_2$ are common to all K components. $(\beta_1 - m_1 + 1)$ and $(\beta_2 - m_2 + 1)$ are given vague Gamma priors, and $\mathbf{C}_1$ and $\mathbf{C}_2$ are given vague Wishart priors:

$$
\begin{aligned}
(\beta_1 - m_1 + 1)^{-1} &\sim \mathcal{G}((\beta_1 - m_1 + 1)^{-1}, 1, 1) && (5.68)\\
(\beta_2 - m_2 + 1)^{-1} &\sim \mathcal{G}((\beta_2 - m_2 + 1)^{-1}, 1, 1) && (5.69)\\
\mathbf{C}_1 &\sim \mathcal{W}(\mathbf{C}_1 \mid m_1, \boldsymbol{\Sigma}_{y_1}) && (5.70)\\
\mathbf{C}_2 &\sim \mathcal{W}(\mathbf{C}_2 \mid m_2, \boldsymbol{\Sigma}_{y_2}) && (5.71)
\end{aligned}
$$

The posterior distribution over $\mathbf{C}_1$ given all K precision matrices is given by:

$$\begin{aligned} p(\mathbf{C}_1 \mid \mathbf{A}_{1,1},..,\mathbf{A}_{1,K},\beta_1) &\propto \prod_{i=1}^{K} p(\mathbf{A}_{1,i} \mid \beta_1, \mathbf{C}_1) p(\mathbf{C}_1) && (5.72) \\ &\propto \prod_{i=1}^{K} \mathcal{W}(\mathbf{A}_{1,i} \mid \beta_1, \mathbf{C}_1) \mathcal{W}(\mathbf{C}_1 \mid m_1, \boldsymbol{\Sigma}_{y_1}) && (5.73) \end{aligned}$$

We can write $\prod_{i=1}^{K} \mathcal{W}(\mathbf{A}_{1,i} \mid \beta_1, \mathbf{C}_1)$ as a function of $\mathbf{C}_1$:

$$\begin{aligned} \prod_{i=1}^{K} \mathcal{W}(\mathbf{A}_{1,i} \mid \beta_1, \mathbf{C}_1) &\propto \prod_{i=1}^{K} \mathcal{W}\left(\mathbf{C}_1 \mid \beta_1 + m_1 + 1, \frac{\beta_1 + m_1 + 1}{\beta_1 \mathbf{A}_{1,i}}\right) && (5.74) \\ &\propto \mathcal{W}\left(\mathbf{C}_1 \mid \beta_1 K + m_1 + 1, \frac{\beta_1 K + m_1 + 1}{\beta_1 \sum_i \mathbf{A}_{1,i}}\right) && (5.75) \end{aligned}$$

Putting this expression back into (5.73), the posterior distribution over $\mathbf{C}_1$ is derived as:

$$\begin{aligned} p(\mathbf{C}_1 \mid \mathbf{A}_{1,1},..,\mathbf{A}_{1,K},\beta_1) &\propto \mathcal{W}\left(\mathbf{C}_1 \mid \beta_1 K + m_1 + 1, \frac{\beta_1 K + m_1 + 1}{\beta_1 \sum_i \mathbf{A}_{1,i}}\right) \\ &\times \mathcal{W}(\mathbf{C}_1 \mid m_1, \boldsymbol{\Sigma}_{y_1}) \\ &\propto \mathcal{W}\left(\mathbf{C}_1 \mid \beta_1 K + m_1, \frac{\beta_1 K + m_1}{m_1 \boldsymbol{\Sigma}_{y_1}^{-1} + \beta_1 \sum_i \mathbf{A}_{1,i}}\right) && (5.76) \end{aligned}$$

Similarly, the posterior distribution over $\mathbf{C}_2$ given the precision matrices $\mathbf{A}_{2,1}, ..., \mathbf{A}_{2,K}$ is given by:

$$p(\mathbf{C}_2 \mid \mathbf{A}_{2,1},..,\mathbf{A}_{2,K},\beta_2) \propto \mathcal{W}\left(\mathbf{C}_2 \mid \beta_2 K + m_2, \frac{\beta_2 K + m_2}{m_2 \boldsymbol{\Sigma}_{y_2}^{-1} + \beta_2 \sum_i \mathbf{A}_{2,i}}\right) \quad (5.77)$$

The posterior distribution over β_1 given all K precision matrices is given by:

$$\begin{aligned} p(\beta_1 \mid \mathbf{A}_{1,1},..,\mathbf{A}_{1,K},\mathbf{C}_1) &\propto \prod_{i=1}^{K} p(\mathbf{A}_{1,i} \mid \beta_1, \mathbf{C}_1) p(\beta_1) && (5.78) \\ &\propto \prod_{i=1}^{K} \mathcal{W}(\mathbf{A}_{1,i} \mid \beta_1, \mathbf{C}_1) \mathcal{G}((\beta_1 - m_1 + 1)^{-1} \mid 1, 1) && (5.79) \end{aligned}$$

and similarly,

$$p(\beta_2 \mid \mathbf{A}_{2,1}, .., \mathbf{A}_{2,K}, \mathbf{C}_2) \;\propto\; \prod_{i=1}^{K} \mathcal{W}(\mathbf{A}_{2,i} \mid \beta_2, \mathbf{C}_2) \mathcal{G}((\beta_2 - m_2 + 1)^{-1} \mid 1, 1) \tag{5.80}$$

Since the latter densities are not of standard form, independent samples are generated from $\log \beta_1 \mid \mathbf{A}_{1,1}, ..., \mathbf{A}_{1,K}\mathbf{C}_1$ and $\log \beta_2 \mid \mathbf{A}_{2,1}, ..., \mathbf{A}_{2,K}, \mathbf{C}_2$ (which can be shown to be log concave distributions) using the Adaptive Rejection Sampling (ARS) technique (Gilks & Wild, 1992).

5.4.3.3 Weight vectors $\mathbf{W}_{1,k}, \mathbf{W}_{2,k}$

The weight matrices for the kth latent variable model are $\mathbf{W}_{1,k}$ and $\mathbf{W}_{2,k}$. The rows of these matrices are drawn from a Gaussian prior such that:

$$\mathbf{W}^i_{1,k} \sim \mathcal{N}(\mathbf{W}^i_{1,k} \mid \boldsymbol{\gamma}_{1,i}, v_1^{-1}\mathbf{I}_q) \tag{5.81}$$

$$\mathbf{W}^i_{2,k} \sim \mathcal{N}(\mathbf{W}^i_{2,k} \mid \boldsymbol{\gamma}_{2,i}, v_2^{-1}\mathbf{I}_q) \tag{5.82}$$

where $\mathbf{W}^i_{1,k}$ and $\mathbf{W}^i_{2,k}$ are the ith rows of $\mathbf{W}_{1,k}$ and $\mathbf{W}_{2,k}$ respectively, $\boldsymbol{\gamma}_{1,i}$ and $\boldsymbol{\gamma}_{1,i}$ are the means of the corresponding distributions, and v_1 and v_2 are the inverse variance. The posterior distribution over $\mathbf{W}^i_{1,k}$ given the data $\mathbf{Y}_1$, the latent variables $\mathbf{X}$, the indicators $\mathbf{c}$, and parameters $\theta^{1,k}$, is given by

$$p(\mathbf{W}^i_{1,k} \mid \mathbf{Y}_1, \mathbf{X}, \mathbf{c}, \boldsymbol{\gamma}_{1,i}, v_1) \propto p(\mathbf{Y}_1 \mid \mathbf{X}, \mathbf{c}, \theta^{1,k}) p(\mathbf{W}^i_{1,k} \mid \boldsymbol{\gamma}_{1,i}, v_1) \tag{5.83}$$

Rewriting $p(\mathbf{Y}_1 \mid \mathbf{X}, \mathbf{c}, \theta^{1,k})$ in terms of $\mathbf{W}^i_{1,k}$ gives:

$$\begin{aligned} p(\mathbf{Y}_1 \mid \mathbf{X}, \mathbf{c}, \theta^{1,k}) &= \prod_{n:c_n=k} \mathcal{N}(\mathbf{y}_{1,n} \mid \mathbf{W}_{1,k}\mathbf{x}_n + \boldsymbol{\mu_{1,k}}, \boldsymbol{\Psi}_{1,k}) && (5.84)\\ &= \prod_{i=1}^{m_1} \mathcal{N}(\mathbf{Y}^i_{1,k} \mid \mathbf{X}_k\mathbf{W}^i_{1,k} + \mu^i_{1,k}, \boldsymbol{\Psi}_{1,k}(i,i)\mathbf{I}_{N_k}) && (5.85)\\ &= \prod_{i=1}^{m_1} \mathcal{N}(\mathbf{W}^i_{1,k} \mid (\mathbf{X}_k^\top\mathbf{X}_k)^{-1}\mathbf{X}_k^\top(\mathbf{Y}^i_{1,k} - \mu^i_{1,k}), \boldsymbol{\Psi}_{1,k}(i,i)\mathbf{X}_k^\top\mathbf{X}_k) && (5.86) \end{aligned}$$

where we have approximated $\boldsymbol{\Psi}_{1,k}$ by its diagonal in (5.85). $\mathbf{Y}^i_{1,k} = \{\mathbf{y}^i_{1,n}\}_{n:c_n=k}$ is the ith dimension of the subset of $\mathbf{Y}_1$ assigned to the kth cluster. $\mathbf{X}_k = \{\mathbf{x}_n\}_{n:c_n=k}$ is the latent variable set associated with the kth cluster, $\mu^i_{1,k}$ is the ith dimension of the mean vector $\mu_{1,k}$, and $\boldsymbol{\Psi}_{1,k}(i,i)$ is the ith diagonal element of $\boldsymbol{\Psi}_{1,k}$. Using this expression with (5.83), the posterior distribution over $\mathbf{W}^i_{1,k}$ becomes

$$\begin{aligned} p(\mathbf{W}^i_{1,k} \mid \mathbf{Y}_1, \mathbf{X}, \mathbf{c}, \boldsymbol{\gamma}_{1,i}, v_1) &\propto \prod_{i=1}^{m_1} \mathcal{N}(\mathbf{W}^i_{1,k} \mid (\mathbf{X}_k^\top\mathbf{X}_k)^{-1}\mathbf{X}_k^\top(\mathbf{Y}^i_{1,k} - \mu^i_{1,k}), \boldsymbol{\Psi}_{1,k}(i,i)\mathbf{X}_k^\top\mathbf{X}_k)\\ &\times\ \mathcal{N}(\mathbf{W}^i_{1,k} \mid \boldsymbol{\gamma}_{1,i}, v_1^{-1}\mathbf{I}_q)\\ &\propto \mathcal{N}(\mathbf{W}^i_{1,k} \mid \mu_{\mathbf{W}^i_{1,k}}, \boldsymbol{\Sigma}_{\mathbf{W}^i_{1,k}}) && (5.87) \end{aligned}$$

where

$$\begin{aligned} \boldsymbol{\Sigma}_{\mathbf{W}^i_{1,k}} &= \left(\boldsymbol{\Psi}_{1,k}(i,i)^{-1}\mathbf{X}_k^\top\mathbf{X}_k + v_1\mathbf{I}_q\right)^{-1}\\ \mu_{\mathbf{W}^i_{1,k}} &= \boldsymbol{\Sigma}_{\mathbf{W}^i_{1,k}}\left((\boldsymbol{\Psi}_{1,k}(i,i))^{-1}\mathbf{X}_k^\top(\mathbf{Y}^i_{1,k} - \mu^i_{1,k}) + v_1\boldsymbol{\gamma}_{1,i}\right) && (5.88) \end{aligned}$$

Similarly, the posterior distribution over $\mathbf{W}^i_{2,k}$ is given by:

$$p(\mathbf{W}^i_{2,k} \mid \mathbf{Y}_2, \mathbf{X}, \mathbf{c}, \boldsymbol{\gamma}_{2,i}, v_2) \propto \mathcal{N}(\mathbf{W}^i_{2,k} \mid \mu_{\mathbf{W}^i_{2,k}}, \boldsymbol{\Sigma}_{\mathbf{W}^i_{2,k}}) \qquad (5.89)$$

where

$$\begin{aligned} \boldsymbol{\Sigma}_{\mathbf{W}^i_{2,k}} &= \left(\boldsymbol{\Psi}_{2,k}(i,i)^{-1}\mathbf{X}_k^\top\mathbf{X}_k + v_2\mathbf{I}_q\right)^{-1}\\ \mu_{\mathbf{W}^i_{2,k}} &= \boldsymbol{\Sigma}_{\mathbf{W}^i_{2,k}}\left((\boldsymbol{\Psi}_{2,k}(i,i))^{-1}\mathbf{X}_k^\top(\mathbf{Y}^i_{2,k} - \mu^i_{2,k}) + v_2\boldsymbol{\gamma}_{2,i}\right) && (5.90) \end{aligned}$$

The hyperparameters are given the following vague priors:

$$v_1 \sim \mathcal{G}(v_1 \mid 1, 1) \tag{5.91}$$

$$v_2 \sim \mathcal{G}(v_2 \mid 1, 1) \tag{5.92}$$

$$\boldsymbol{\gamma}_{1,i} \sim \mathcal{N}(\boldsymbol{\gamma}_{1,i} \mid 0, \tau^{-1}\mathbf{I}_q) \tag{5.93}$$

$$\boldsymbol{\gamma}_{2,i} \sim \mathcal{N}(\boldsymbol{\gamma}_{2,i} \mid 0, \tau^{-1}\mathbf{I}_q) \tag{5.94}$$

The posterior distribution over the inverse scale hyperparameter v_1 is given by

$$\begin{aligned} p(v_1 \mid \mathbf{W}_{1,1}, ..., \mathbf{W}_{1,K}, \boldsymbol{\gamma}_{1,i}) &\propto \prod_{k=1}^{K} p(\mathbf{W}_{1,k} \mid v_1, \boldsymbol{\gamma}_1) p(v_1) \\ &\propto \prod_{i=1}^{m_1} \prod_{k=1}^{K} \mathcal{N}(\mathbf{W}^i_{1,k} \mid \boldsymbol{\gamma}_{1,i}, v_1^{-1}\mathbf{I}_q) \mathcal{G}(v_1 \mid 1, 1) \\ &\propto \mathcal{G}\left(v_1 \mid m_1 K + 1, \frac{m_1 K + 1}{1 + \mathbf{S}_{W_1}}\right) \end{aligned} \tag{5.95}$$

where $\mathbf{S}_{W_1} = \sum_{i=1}^{m_1} \sum_k (\mathbf{W}^i_{1,k} - \boldsymbol{\gamma}_{1,i})^\top (\mathbf{W}^i_{1,k} - \boldsymbol{\gamma}_{1,i})$. Similarly, the posterior distribution over v_2 is given by:

$$p(v_2 \mid \mathbf{W}_{2,1}, ..., \mathbf{W}_{2,K}, \boldsymbol{\gamma}_{2,i}) \propto \mathcal{G}\left(v_2 \mid m_2 K + 1, \frac{m_2 K + 1}{1 + \mathbf{S}_{W_2}}\right) \tag{5.96}$$

where $\mathbf{S}_{W_2} = \sum_{i=1}^{m_2} \sum_k (\mathbf{W}^i_{2,k} - \boldsymbol{\gamma}_{2,i})^\top (\mathbf{W}^i_{2,k} - \boldsymbol{\gamma}_{2,i})$. The posterior distribution over the hyperparameter $\boldsymbol{\gamma}_{1,i}$ is given by:

$$\begin{aligned} p(\boldsymbol{\gamma}_{1,i} \mid \{(\mathbf{W}_{1,k})^i\}_{k=1}^K, v_1) &\propto \prod_{k=1}^{K} p(\mathbf{W}^i_{1,k} \mid v_1, \boldsymbol{\gamma}_1) p(\boldsymbol{\gamma}_{1,i}) \\ &\propto \prod_{k=1}^{K} \mathcal{N}(\mathbf{W}^i_{1,k} \mid \boldsymbol{\gamma}_{1,i}, v_1^{-1}\mathbf{I}_q) \mathcal{N}(\boldsymbol{\gamma}_{1,i} \mid 0, \tau^{-1}\mathbf{I}_q) \\ &\propto \mathcal{N}\left(\boldsymbol{\gamma}_{1,i} \mid \frac{v_1 \sum_k \mathbf{W}^i_{1,k}}{K v_1 + \tau}, \frac{1}{K v_1 + \tau}\right) \end{aligned} \tag{5.97}$$

Similarly, the posterior over $\boldsymbol{\gamma}_{2,i}$ is given by:

$$p(\boldsymbol{\gamma}_{2,i} \mid \{\mathbf{W}_{2,k}^{i}\}_{k=1}^{K}, v_2) \quad \propto \quad \mathcal{N}\left(\boldsymbol{\gamma}_{2,i} \mid \frac{v_2 \sum_k \mathbf{W}_{2,k}^{i}}{Kv_2 + \tau}, \frac{1}{Kv_2 + \tau}\right) \tag{5.98}$$

5.4.3.4 Latent variable x

The latent variable $\mathbf{x}_n$ for the nth pair of data points $\mathbf{y}_n = [\mathbf{y}_{1,n}^{\top}, \mathbf{y}_{2,n}^{\top}]^{\top}$ is drawn from a Gaussian prior with zero mean and unit variance:

$$\mathbf{x}_n \sim \mathcal{N}(\mathbf{x}_n \mid 0, \mathbf{I}_q) \tag{5.99}$$

The posterior distribution over $\mathbf{x}_n$ is given by:

$$\begin{aligned} p(\mathbf{x}_n \mid \theta, \mathbf{y}_{1,n}, \mathbf{y}_{2,n}) \quad &\propto \quad p(\mathbf{y}_{1,n}, \mathbf{y}_{2,n} \mid \mathbf{x}_n, \theta) p(\mathbf{x}_n) \\ &\propto \quad \mathcal{N}(\mathbf{x}_n \mid \mu_{\mathbf{x}_n}, \boldsymbol{\Sigma}_{\mathbf{x}_n}) \qquad (5.100) \end{aligned}$$

where

$$\mu_{\mathbf{x}_n} \quad = \quad \mathbf{W}_{c_n}^{\top}(\mathbf{W}_{c_n}\mathbf{W}_{c_n}^{\top} + \boldsymbol{\Psi}_{c_n})^{-1}(\mathbf{y}_n - \mu_{c_n}) \tag{5.101}$$

$$\boldsymbol{\Sigma}_{\mathbf{x}_n} \quad = \quad \mathbf{I}_q - \mathbf{W}_{c_n}^{\top}(\mathbf{W}_{c_n}\mathbf{W}_{c_n}^{\top} + \boldsymbol{\Psi}_{c_n})^{-1}\mathbf{W}_{c_n} \tag{5.102}$$

where $c_n \in \{1, ..., K\}$ denotes the component index which generated $\mathbf{y}_n$, $\mathbf{W}_{c_n}, \mu_{c_n}$ and $\boldsymbol{\Psi}_{c_n}$ are the parameters of the corresponding component, with $\mathbf{W}_{c_n} = [\mathbf{W}_{1,c_n}^{\top}\mathbf{W}_{2,c_n}^{\top}]^{\top}, \mu_{c_n} = [\mu_{1,c_1}^{\top}\mu_{2,c_1}^{\top}]^{\top}$, and $\boldsymbol{\Psi}_{c_n} = \begin{pmatrix} \boldsymbol{\Psi}_{1,c_n} & 0 \\ 0 & \boldsymbol{\Psi}_{2,c_n} \end{pmatrix}$

5.4.3.5 Indicators c_n

The conditional priors on the indicators is given by:

$$c_n = k \mid \mathbf{c}_{-n}, \alpha_0 \quad = \quad \frac{N_{-n,k}}{N - 1 + \alpha_0} \tag{5.103}$$

$$c_n \neq c_{n'} \forall n' \neq n \mid \mathbf{c}_{-n}, \alpha \quad = \quad \frac{\alpha_0}{N - 1 + \alpha_0} \tag{5.104}$$

where $-n$ indicates all the indices except n, such that $\mathbf{c}_{-n}$ denotes all the indicators except the nth, and $N_{-n,k}$ is the number of data points associated with the kth component, excluding the nth data point. The first equation shows that the conditional prior probability of the nth data point being assigned to the kth component, given the assignments of the other data points, is proportional to the number of data points in the kth cluster. The second equation shows that the combined prior for the nth data point being assigned to one of the infinite unrepresented classes is only dependent on α_0 and N. α_0 is the concentration parameter, and controls the amount of 'left over' probability mass corresponding to data being assigned to the currently unrepresented classes. A vague Gamma prior is placed over α_0:

$$\alpha_0 \sim \mathcal{G}(\alpha_0 \mid 1, 1) \tag{5.105}$$

The posterior distributions over the indicators is given by the following: for components for which $N_{-n,k} > 0$

$$\begin{aligned} &c_n = k \mid \mathbf{c}_{-n}, \theta_k, \alpha_0 \propto \\ &\tfrac{N_{-n,k}}{N-1+\alpha_0} \mathcal{N}(\mathbf{y}_n \mid \mathbf{W}_k \mathbf{x}_n + \mu_k, \mathbf{\Psi}_k) \end{aligned} \tag{5.106}$$

for all other components:

$$\begin{aligned} &c_n \neq c_{n'} \forall n' \neq n \mid \mathbf{c}_{-n}, \gamma, \alpha_0 \propto \\ &\tfrac{\alpha_0}{N-1+\alpha_0} \int p(\mathbf{y}_n \mid \mathbf{x}_n, \theta^k) p(\mathbf{x}_n) p(\theta^k \mid \gamma) d\mathbf{x}_n d\theta^k \end{aligned} \tag{5.107}$$

The likelihood for currently unrepresented classes (which have no parameters associated with them) is found by integrating over the parameter priors. The posterior distribution over α_0 is given by:

$$\alpha_0 \mid K, N \propto \frac{\alpha_0^K \Gamma(\alpha_0)}{\Gamma(\alpha_0 + N)} \mathcal{G}(\alpha_0 \mid 1, 1) \tag{5.108}$$

This only depends on the number of observations N and the number of represented components K, and not on how the observations are distributed among the components. Samples are generated from $p(\log(\alpha_0) \mid K, N)$, which is log concave, using ARS.

5.4.4 Inference in the model

As noted before, exact analytical inference is not possible in this model, and Gibbs sampling is used to update the parameters, hyperparameters and indicator variables. Each variable in turn is updated by sampling from its posterior distribution conditional on all the other variables as follows:

- The parameters are updated by sampling from $p(\theta \mid \gamma, \mathbf{c}, \mathbf{Y})$
- The hyperparameters are updated by sampling from $p(\gamma \mid \theta, \mathbf{c}, \mathbf{Y})$
- The indicator variables are updated by sampling from $p(\mathbf{c} \mid \theta, \gamma, \mathbf{Y})$
- The concentration parameter is updated by sampling from $p(\log(\alpha_0) \mid K, N)$

This process (a Gibbs sweep) generates a sample from the joint posterior distribution $p(\theta, \gamma, \mathbf{c} \mid \mathbf{Y})$. Many Gibbs sweeps are performed to repeatedly update all the variables. Since consecutive samples are likely to be correlated, in order to generate independent samples from the joint posterior, the mixing time of the Markov chain is calculated and a sample is taken in every period of this length.

5.5 Experiments

To illustrate the model, we use a pair of toy data sets (each 2 dimensional) where the first data set follows an arc, and the second data set follows a sine curve.

To perform inference for the model, we initialise the model with one component and then perform a large number of Gibbs sweeps to update the hyperparameters, parameters, and indicator variables, storing the values at each iteration. Initially, we do not know how the Markov chain will mix and converge for this particular data set so we perform 10000 iterations to assess the mixing and convergence times. Figure 5.7 shows the number of represented components K plotted for each Monte Carlo iteration. K

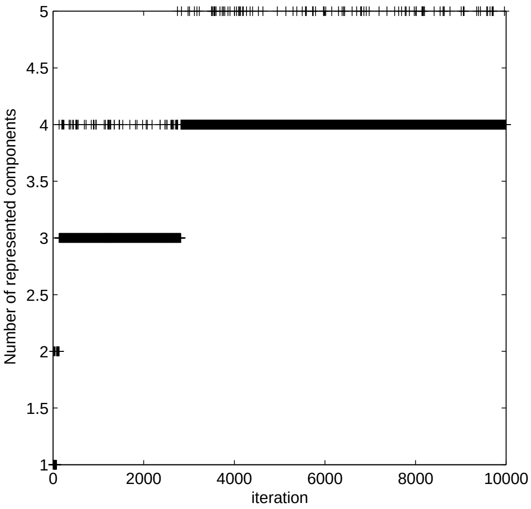


Figure 5.7: The number of represented components K growing with each Monte Carlo iteration. The burn-in time is estimated to be 3000 iterations.

grows with time and the convergence time (or the burn-in time) is approximately 3000 iterations. Discarding the 3000 iterations produced during the burn-in phase, the mixing time for the Markov chain is estimated by plotting the autocovariance for different parameters against time (based on 10000 iterations) and finding the maximum correlation length. The autocovariance against lag plot is shown in Figure 5.8, and it can be seen that there are no significant correlations for any of the parameters. We choose the effective correlation length to be 10 iterations.

We then perform 10000 iterations for modelling purposes - 3000 for the burn-in period, and a further 7000 which generates 700 independent samples from the posterior distribution (spaced evenly 10 apart). Figure 5.9 shows four sets of samples from the posterior distribution for the mixture models at iterations 1, 500, 4000, and 6000 (from the 10000 iterations). During the burn-in period, as shown in iterations 1 and 500, the model underfits the data. As more samples are drawn and the Markov chain converges, the model finds that 4 mixture components are the best fit for the data. There is a small amount of probability mass (controlled by α_0) which allows the model to consider an additional component (at iteration 4000). As there is not enough evidence for this component provided by the data, it is removed in the next Gibbs sweep. Figure 5.10 shows the histograms for some parameters of the mixture model, based on the 700

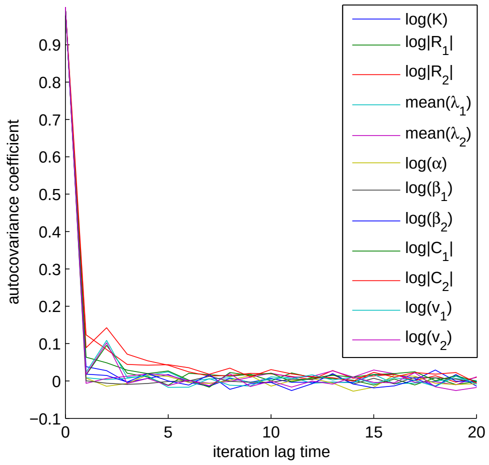


Figure 5.8: The autocovariance plotted against lag, based on 10000 iterations, for various parameters of the mixture model. The effective correlation length is chosen to be 10 iterations.

independent samples from the posterior distribution.

5.5.1 Examining the distribution over the latent space

In the mixture model, there is a set of latent variables $\mathbf{X}$ that underlies both data spaces $\mathbf{Y}_1$ and $\mathbf{Y}_2$. In this section, we find the distribution over $\mathbf{X}$ given just one of the data sets. This distribution can then be used to predict one data set given the other, and vice versa. The posterior distribution over the nth latent variable $\mathbf{x}_n$ given the corresponding data point from the first data set, is given by:

$$p(\mathbf{x}_n \mid \mathbf{y}_{1,n}) = \int p(\mathbf{x}_n \mid \mathbf{y}_{1,n}, \theta) p(\theta) d\theta \tag{5.109}$$

$$= \frac{1}{I} \sum_{i=1}^{I} p(\mathbf{x}_n \mid \mathbf{y}_{1,n}, \theta^i) \tag{5.110}$$

$$= \frac{1}{I} \sum_{i=1}^{I} \mathcal{N}(\mathbf{x}_n \mid (\mu_{\mathbf{x}_n})^i, (\mathbf{\Sigma}_{\mathbf{x}_n})^i) \tag{5.111}$$

where

$$(\mu_{\mathbf{x}_n|\mathbf{y}_{1,n}})^i = (\mathbf{W}_{1,c_n})^{i\top}((\mathbf{W}_{1,c_n})^i(\mathbf{W}_{1,c_n})^{i\top} + (\mathbf{\Psi}_{1,c_n})^i)^{-1}(\mathbf{y}_{1,n} - (\mu_{1,c_n})^i)$$

$$(\mathbf{\Sigma}_{\mathbf{x}_n|\mathbf{y}_{1,n}})^i = \mathbf{I}_q - (\mathbf{W}_{1,c_n})^{i\top}((\mathbf{W}_{1,c_n})^i(\mathbf{W}_{1,c_n})^{i\top} + (\mathbf{\Psi}_{1,c_n})^i)^{-1}(\mathbf{W}_{1,c_n})^i$$

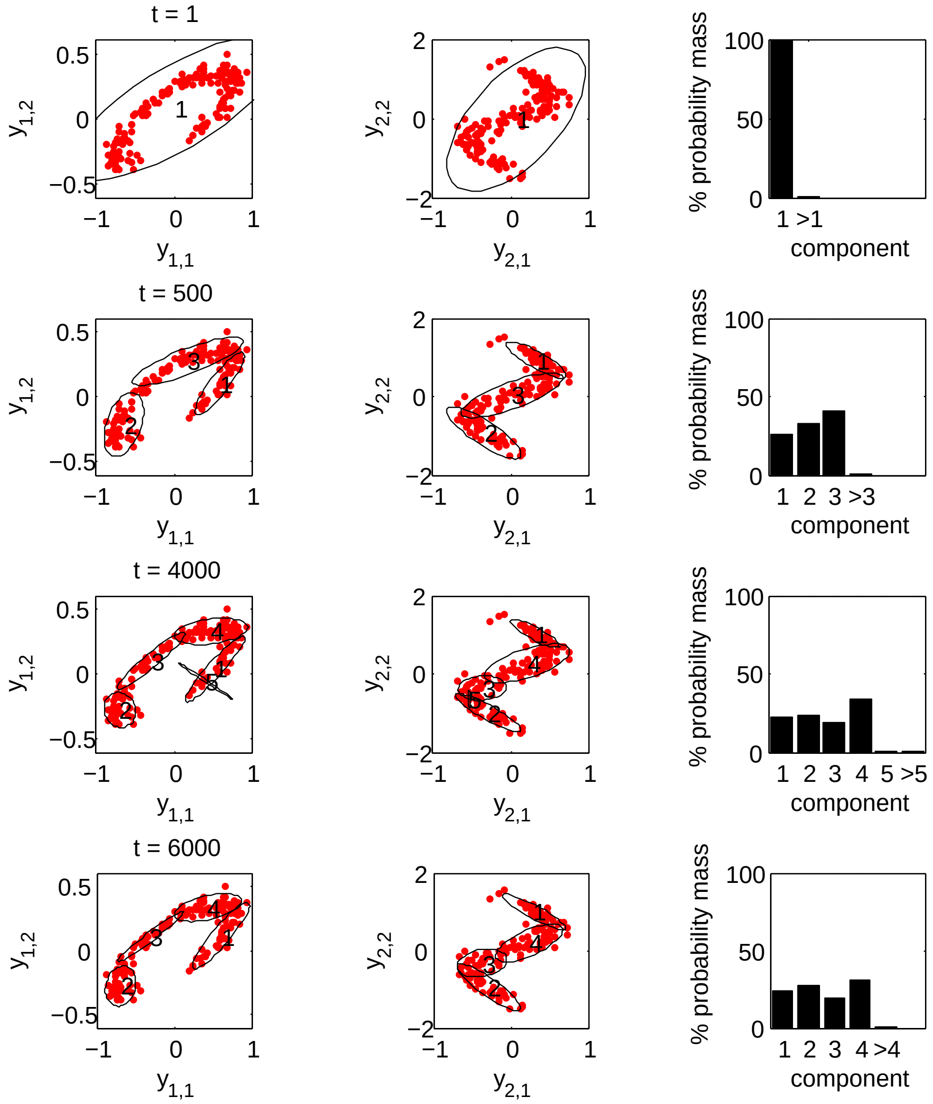


Figure 5.9: Four sets of samples from the posterior distribution for the mixture model, at iterations 1, 500, 4000, and 6000. Each row shows a sample over the first data set $\mathbf{Y}_1$ (first column) and the second data set $\mathbf{Y}_2$ (second column), and a graph for the probability mass in each component and the unrepresented components (third column). The ellipses indicate 2 standard deviations of the noise covariance matrices of each component, and the labels for each component 1,...,K are positioned at the means.

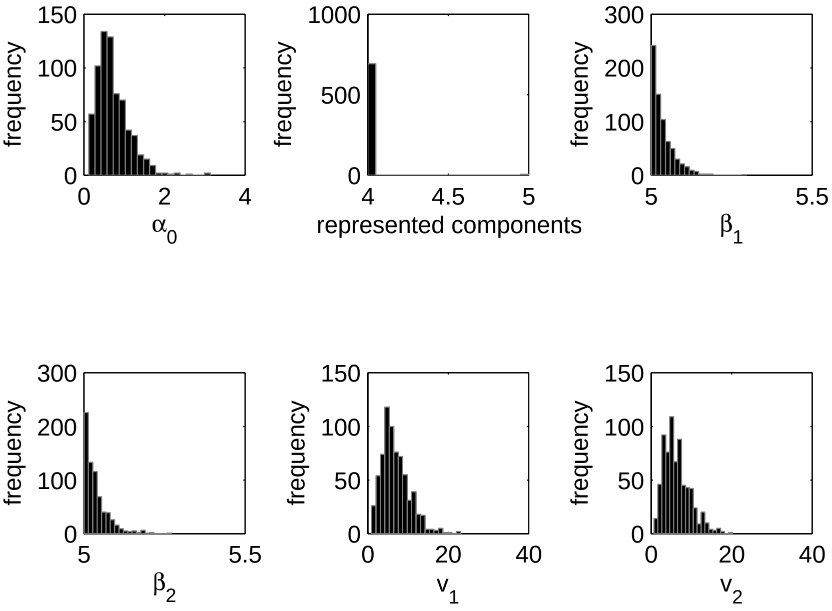


Figure 5.10: Some histograms for the posterior over different parameters in the model, given the data, based on 700 independent samples from the posterior

where I is the number of independent samples, and the superscript i denotes the ith independent sample, such that θ^i describes the ith sample of the posterior over θ.

5.5.2 Predictive distribution

After finding the posterior distribution over the latent space given one data set, we can evaluate the predictive distribution over the other data space, according to:

$$p(\mathbf{y}_{2,n} \mid \mathbf{y}_{1,n}) = \int p(\mathbf{y}_{2,n} \mid \mathbf{x}_n, \theta) p(\mathbf{x}_n \mid \mathbf{y}_{1,n}) p(\theta) d\mathbf{x}_n d\theta \qquad (5.112)$$

$$= \frac{1}{I} \sum_{i=1}^{I} \int p(\mathbf{y}_{2,n} \mid \mathbf{x}_n, \theta_i) p(\mathbf{x}_n \mid \mathbf{y}_{1,n}) d\mathbf{x}_n \qquad (5.113)$$

Figure 5.11 shows the predictive distribution over each data set given the other. As can be seen from the figure, the model is able to infer the distribution over the nonlinear manifold underlying each data set given the other, using an appropriate number of mixture components.

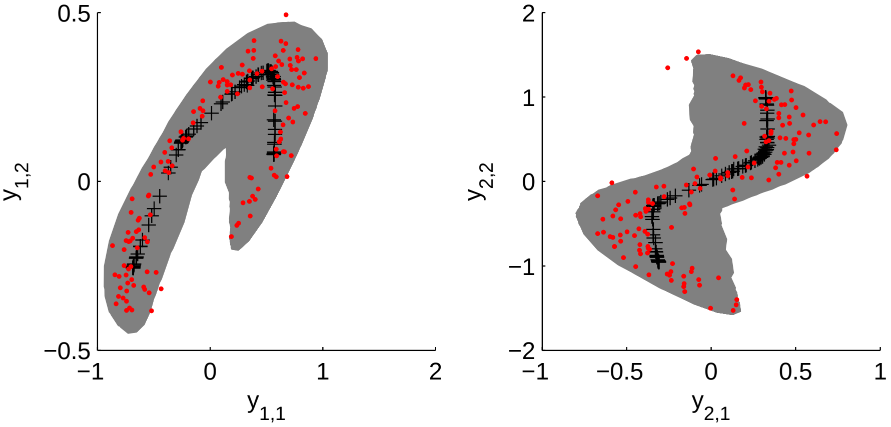


Figure 5.11: The predictive distribution over each data set given the other. The predictive mean is shown in black, 2 standard deviations of the predictive variance in grey, and the data is shown in red

5.6 Conclusion

In this chapter, we presented a model for finding a joint probabilistic representation of two data sources, where each data source lies close to a nonlinear manifold embedded in the data space, each indexed by a shared set of latent coordinates. One of the problems of defining nonlinear mappings between the latent and data spaces is that a unique solution does not exist, and the mappings have to be constrained appropriately so that they do not underfit or overfit the data. When approximating a nonlinear manifold with a mixture of local linear latent variable models, inferring the correct model complexity from the data is an important issue; we have to use an appropriate number of mixture components since this governs the flexibility of the manifold. Additionally, since we want to model two nonlinear manifolds, the difficulty of the problem is increased. Unfortunately, when using maximum likelihood methods, as is the standard procedure for mixture models, we cannot infer the number of mixture components for the model.

We considered a mixture model of probabilistic canonical correlation analysers such that the pair of nonlinear manifolds was approximated by local linear submodels; at corresponding local regions of the two data spaces, the relationship between the data was modelled by a local PCCA model. To address the model selection problem, we used nonparametric Bayesian techniques which allowed the data to determine the necessary complexity of the model.

A nonparametric Dirichlet process prior was placed over the parameters of the mixture model of PCCA. This allowed the number of represented mixture components, and hence the flexibility of the nonlinear manifolds, to be determined automatically. We call this model a Dirichlet process mixture of PCCA. The data is modelled as being generating from an infinite mixture of PCCA, in the same spirit as the infinite mixture of Gaussians (Rasmussen, 2000).

We demonstrated the model on a toy problem, and found that the model was able to correctly infer the necessary number of mixture components to represent the relationship between the data spaces.

Chapter 6

Conclusion

6.1 Discussion

6.1.1 Probabilistic generative approach to context assisted learning

In this thesis, we have presented a probabilistic generative framework for analysing two sets of data, where the underlying structure to each data set is learned by taking its context (the other data set) into account. We represent the structure of each data set as the sum of a shared function and a private, or noise function. The two shared functions are related through a common latent variable which forms a low dimensional representation, or embedding, of the relationship between the two data sets. The relationship between the two sets of data variables $\mathbf{y}_1$ and $\mathbf{y}_2$ is described probabilistically in terms of the shared structure in the latent variable $\mathbf{x}$, and the noise processes. After learning the shared structure of the model, we can then manipulate the joint probability density over the variables to calculate such quantities as the predictive densities $p(\mathbf{y}_1 \mid \mathbf{y}_2)$ and $p(\mathbf{y}_2 \mid \mathbf{y}_1)$, or $p(\mathbf{x} \mid \mathbf{y}_1)$ and $p(\mathbf{x} \mid \mathbf{y}_2)$, the posterior distributions over the latent space (the representation of the data in the feature space). The advantage of the dependency-seeking models that we describe in this thesis is that they are fully probabilistic, and that they could also be generalised to multiple data sets. The models can be interpreted as probabilistic nonlinear canonical correlation analysis models.

6.1.2 Nonparametric Bayesian methods and probabilistic nonlinear CCA

We are interested in modelling two data sets that have a complex relationship. We model each data set as lying close to a nonlinear manifold indexed by a shared set of latent coordinates i.e. we assume that we can represent each data set as a set of low dimensional features that are nonlinearly related to the data space. One of the problems of modelling nonlinear structure is that there is an indeterminacy in the solution, and it is necessary to appropriately constrain the mappings such that the model does not overfit or underfit the data. Additionally, since we require two nonlinear mappings, the complexity of the problem is increased. If the mappings are too flexible, then the model may find spurious correlations between the data sets. If the mappings are too inflexible, then the model may not find the underlying shared structure between the data sets.

One approach to specifying nonlinear functions is to *a priori* define the form of the function whose complexity is controlled by a finite set of parameters. Learning the function then consists of finding the best setting of the parameters from the data by maximum likelihood or maximum a posteriori methods. However, the problem of inferring model complexity still remains since the structure of the model is set *a priori* and is not learned during the optimisation. In this thesis, we used nonparametric Bayesian methods to overcome the problem of modelling nonlinear structure. Nonparametric Bayesian methods can be used to place flexible priors over models, such that the model complexity is automatically inferred from the data set and can adapt to new data points.

In Chapters 3 and 4, we used Gaussian processes as a prior over the functions from latent to data spaces. This does not restrict the class of possible functions, as in parametric modelling. By placing a prior over the space of functions, and giving higher probability to functions that have the desired characteristics (e.g. smoothness), this allows a rich class of possible functions to be considered within a principled framework. In Chapter 5, we approximated the nonlinear manifolds underlying the data by a mixture of probabilistic canonical correlation analysers. The problem of constraining the mapping arises in setting the number of mixture components for the model. We resolve this problem by using a nonparametric Dirichlet process (DP) prior on the parameters for each pair of data points. When drawing the parameter set from the DP prior, there is a clustering effect i.e. the draws from the priors are not necessarily distinct, and may take on values of previous draws. Pairs of data points that share the same setting of the parameters can be viewed as belonging to the same mixture component (or cluster). This does not require the number of mixture components to be set in advance. Instead the model considers an infinite number of mixture components, where the number of *represented* components are determined by the data. This allows the necessary complexity of the mappings between latent and data spaces to be determined automatically.

6.2 Review of the thesis

Chapter 2 provided a background to the work in this thesis, and Chapters 3, 4 and 5 provided the new work in the thesis. In Chapter 3 we introduced a Gaussian process latent variable model of canonical correlation analysis (GPLVM-CCA). We then showed that the within-set variation in two related data sets could be modelled by using linear transformations $\mathbf{\Psi}_1^{-1/2}$ and $\mathbf{\Psi}_2^{-1/2}$ of each data set, and showed that the generative dependency seeking model, probabilistic canonical correlation analysis (Bach & Jordan, 2005), could be interpreted within this framework. We then extended this model in the spirit of the Gaussian process latent variable model (GPLVM) (Lawrence, 2004) to model two related data sets. Gaussian process (GP) priors are placed over each dimensions of each data set. The covariance functions for each data set define an implicit nonlinear mapping from the latent space to the data space. The shared information is captured in a shared set of latent coordinates (which are the input to the GP's), and the private information is captured in the linear transformations $\mathbf{\Psi}_1$ and $\mathbf{\Psi}_2$, which is automatically learned in the training of the model. The model was applied to various problems where we show that the model can learn an appropriate shared structure between two related data sets when the features are both linearly and nonlinearly related to the data sets. We also demonstrated the algorithm on a pair of large data sets, where each pair of data points consists of a left and right half of a face under various poses and expressions. The model was able to learn a shared latent space that reflects the different poses in the data set. Since the model defines a joint probability density for the data sets, we also demonstrate the model on prediction and missing value problems.

Chapter 4 extended the GPLVM-CCA model with a more complex noise process. We created additional latent spaces which underlie the noise processes in each data sets to model structure in the within-set variation. We placed GP priors on the noise functions and optimised the GP's inputs, such that the noise information was modelled by a covariance function with an input private to each data set. We illustrated this model on a standard artificial data set to demonstrate parts-based decompositions of images. Each image contains a shared feature (a horizontal bar) and a private feature (a vertical bar from either the left or right half of the image). Given a large training set of images, the model was able to find a smaller basis of prototype images containing both the shared and private features.

Finally, in Chapter 5, we presented a Dirichlet process mixture model of probabilistic CCA (PCCA). The pair of underlying nonlinear manifolds for each data set is approximated by local linear submodels; at corresponding local regions of the two data spaces, the relationship between the data is modelled by a local PCCA model. A nonparametric Dirichlet process prior is placed over the parameters of the mixture model of PCCA. This allows the number of represented mixture components, and hence the flexibility of the nonlinear manifolds, to be determined automatically. The data is modelled as being generated from an infinite mixture of PCCA, in the same spirit as the infinite mixture of Gaussians (Rasmussen, 2000).

6.3 Future work

There are a number of avenues for future research:

- Sparse approximations. One of the problems with using nonparametric Bayesian methods is that they can be inefficient in terms of the computation time, since the computing memory required scales prohibitively with the number of data points.

For instance, using Gaussian processes involves manipulation of an $N \times N$ covariance function matrix which is impractical for large data sets. In Chapter 3, we used the informative vector machine (Lawrence *et al.*, 2003; Lawrence, 2004) to create a sparse approximation of GPLVM-CCA. An interesting area of research would be to investigate different sparse approximations for the model since there are many sparsification algorithms in the literature i.e. (Csató, 2002) . Creating a sparse version of the GPLVM-CCA model with complex noise processes (Chapter 4) would also be an interesting area of research. However, there exists a number of problems in this approach due to the richness of the model, such as what criterion would be used to create a sparse version of the covariance function matrix - the accuracy of the noise process or the shared process?

Similarly, the Monte Carlo methods associated with Dirichlet process inference can be computationally costly. A future direction of research would be to use a variational approximation to the inference, as in (Blei & Jordan, 2006).

- Non-Gaussian Processes. A future direction of research would to investigate non-Gaussian noise models for all of our three models.

- Extension of our models to find shared structure for more than two related data sets, following ideas from (Kettenring, 1971).

- Complex structure between two data sets. Since all of our models are probabilistic, it may be interesting to include more prior knowledge about the underlying latent processes into the model. For instance, we could assume that the shared latent variable follows a Markov process (perhaps incorporating dynamics as in (Wang, 2005; Wang *et al.*, 2006)) to model stereo audio data. It would also be interesting to model stereo image data which is a complex problem, and use priors over the latent variables that reflect the data generation process.

Appendix A

Probability distributions

A.1 Normal distribution

$\mathbf{x}$ is a D-dimensional vector distributed according to:

$$\begin{aligned} \mathbf{x} &\sim \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) && \text{(A.1)} \\ &\sim \frac{1}{(2\pi)^{\frac{D}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^{\top}\right)\right) && \text{(A.2)} \end{aligned}$$

where $\boldsymbol{\mu} \in \Re^{D}$ is the mean, and $\boldsymbol{\Sigma} \in \Re^{D \times D}$ is the covariance matrix.

A.1.1 Product of normal distributions

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \propto \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3) \tag{A.3}$$

where

$$\begin{aligned} \boldsymbol{\mu}_3 &= \boldsymbol{\Sigma_3}({\boldsymbol{\Sigma}_1}^{-1}\boldsymbol{\mu}_1 + {\boldsymbol{\Sigma}_2}^{-1}\boldsymbol{\mu}_2) && \text{(A.4)} \\ \boldsymbol{\Sigma}_3 &= ({\boldsymbol{\Sigma}_1}^{-1} + {\boldsymbol{\Sigma}_2}^{-1})^{-1} && \text{(A.5)} \end{aligned}$$

A.2 Gamma distribution

$\mathbf{x}$ is a scalar distributed according to:

$$\begin{aligned}\mathbf{x} &\sim \mathcal{G}(\mathbf{x} \mid \alpha, \beta) && \text{(A.6)}\\ &\sim \frac{\beta^{-\alpha/2}}{\Gamma(\alpha/2)}\mathbf{x}^{\alpha/2-1}\exp(-\alpha\mathbf{x}/2\beta) && \text{(A.7)}\end{aligned}$$

where α is the shape parameter, and β is the mean.

A.3 Wishart distribution

$\mathbf{X}$ is a $D \times D$ matrix distributed according to:

$$\begin{aligned}\mathbf{X} &\sim \mathcal{W}(\mathbf{X} \mid v, \mathbf{S}) && \text{(A.8)}\\ &\sim \frac{1}{Z_{v\mathbf{S}}}|\mathbf{X}|^{(v-D-1)/2}\exp\left(-\frac{1}{2}\mathrm{tr}(v\mathbf{S}^{-1}\mathbf{X})\right) && \text{(A.9)}\end{aligned}$$

where the normalisation constant $Z_{v\mathbf{S}} = 2^{vD/2}\pi^{D(D-1)/4}\frac{|\mathbf{S}|^{v/2}}{v}\prod_{i=1}^{D}\Gamma\left(\frac{v+1-i}{2}\right)$, with degree of freedom v and mean $\mathbf{S}$.

A.3.1 Product of Wishart distributions

$$\mathcal{W}(\mathbf{X} \mid v_3, \mathbf{S}_3) \propto \mathcal{W}(\mathbf{X} \mid v_1, \mathbf{S}_1)\mathcal{W}(\mathbf{X} \mid v_2, \mathbf{S}_2) \tag{A.10}$$

$$\text{where} \tag{A.11}$$

$$v_3 = \sum_i (v_i - D - 1) + D + 1 \tag{A.12}$$

$$\mathbf{S}_3 = \frac{v_3}{v_1\mathbf{S}_1^{-1} + v_2\mathbf{S}_2^{-1}} \tag{A.13}$$

Bibliography

AGAKOV, F. V. 2005. *Variational Information Maximization in Stochastic Environments*. Ph.D. thesis, Institute of Adaptive and Neural Computation, School of Informatics, University of Edinburgh.

AGAKOV, F. V., & BARBER, D. 2004. Variational Information Maximization for Neural Coding. *Pages 543–548 of:* PAL, N. R., KASABOV, N., MUDI, R. K., PAL, S., & PARUI, S. K. (eds), *ICONIP*. Lecture Notes in Computer Science, vol. 3316. Springer.

AKAHO, S., KIUCHI, Y., & UMEYAMA, S. 1999. MICA: Multimodal Independent Component Analysis. *International Joint Conference on Neural Networks (IJCNN)*, **2**, 927–932.

ALDOUS, D. 1985. Exchangeability and Related Topics. *Pages 1–198 of: Ecole D'Ete de Probabilities de Saint Flour XIII 1983*. Springer, Berlin.

ANTONIAK, C. E. 1974. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Annals of Statistics*, **2**(6), 1152–1174.

ARCHAMBEAU, C., DELANNAY, N., & VERLEYSEN, M. 2006. Robust Probabilistic Projections. *Pages 33–40 of:* COHEN, W.W., & MOORE, A. (eds), *Proceedings of the 23rd International Conference on Machine Learning*.

BACH, F.R., & JORDAN, M.I. 2002. Kernel Independent Component Analysis. *Journal of Machine Learning*, **3**, 1–48.

BACH, F.R., & JORDAN, M.I. 2005. *A Probabilistic Interpretation of Canonical Correlation Analysis*. Tech. rept. 688. Dept of Statistics, University of California.

BARTHOLOMEW, D. J. 1987. *Latent Variable Models and Factor Analysis*. London: Charles Griffin and Co. Ltd.

BECKER, S. 1992. *An Information-theoretic Unsupervised Learning Algorithm for Neural Networks*. Ph.D. thesis, University of Toronto.

BECKER, S. 1996. Mutual Information Maximization: models of cortical self-organisation. *Network: Computation in Neural Systems*, **7**, 7–31.

BECKER, S., & HINTON, G. E. 1992. A self-organising neural network that discovers surfaces in random-dot stereograms. *Nature*, **365**(353), 161–163.

BISHOP, C.M. 1999. Latent Variable Models. *Pages 371–403 of:* JORDAN, M. I. (ed), *Learning in Graphical Models*. MIT Press.

BISHOP, C.M., SVENSÉN, M., & WILLIAMS, C.K.I. 1996. GTM: A principled alternative to the Self Organising Map. *Pages 354–360 of:* MOZER, M.C., JORDAN, M.I., & T.PETCHE (eds), *Advances in Neural Information Processing Systems*, vol. 9.

BISHOP, C.M., SVENSÉN, M., & WILLIAMS, C.K.I. 1998. Developments of the Generative Topographic Mapping. *Neurocomputing*, **21**, 203–224.

BLACKWELL, D., & MACQUEEN, J. B. 1973. Ferguson Distributions via Pólya Urn Schemes. *Annals of Statistics*, **1**(2), 353–355.

BLEI, D. M., & JORDAN, M. I. 2006. Variational Inference for Dirichet Process Mixtures. *Bayesian Analysis,* **1**, 121–144.

BORGA, M. 1998. *Learning multidimensional signal processing*. Ph.D. thesis, Linköping University, Sweden, SE-583 83 Linköping, Sweden, Dissertation No. 531, ISBN 91-7219-202-X.

BOYLE, P., & FREAN, M. 2005a. Dependent Gaussian Processes. *Pages 217 –224 of:* SAUL, L. K., WEISS, Y., & BOTTOU, L. (eds), *Advances in Neural Information Processing Systems*, vol. 17. MIT Press.

BOYLE, P., & FREAN, M. 2005b. *Multiple-Output Gaussian Process Regression*. Tech. rept. CS-TR-05/2. School of Mathematical and Computing Science, Victoria University of Wellington.

BREGLER, C., & OMOHUNDRO, S. M. 1995. Nonlinear image interpolation using manifold learning. *Pages 973–980 of:* TESAURO, G., TOURETZKY, D. S., & LEEN, T. K. (eds), *Advances in Neural Information Processing Systems*, vol. 7. MIT Press.

BUTZ, T., & THIRAN, J.P. 2005. From error probability to information theoretic (multi-modal) signal processing. *Signal Processing*, **85**(5), 875–902.

CHARLES, D., & FYFE, C. 1998. Modelling multiple cause structure using rectification constraints. *Network: Computation in Neural Systems*, **9**(2), 167–82.

CHECHIK, G., & GLOBERSON, A. 2003. *Information Bottleneck and linear projections of Gaussian processes*. Tech. rept. 4. Hebrew University.

CHECHIK, G., GLOBERSON, A., TISHBY, N., & WEISS, Y. 2003. Information Bottleneck for Gaussian variables. *Pages 1213–1220 of:* THRUN, S., SAUL, L.K., & SCHÖLKOPF, B. (eds), *Advances in Neural Information Processing Systems*, vol. 16.

CORDUNEANU, A., & BISHOP, C. M. 2001. Variational Bayesian Model Selection for Mixture Distributions. *Pages 27–34 of:* JAAKKOLA, T., & RICHARDSON, T. (eds), *Artificial Intelligence and Statistics*. Morgan Kaufmann.

CSATÓ, L. 2002. *Gaussian Processes - Iterative Sparse Approximations*. Ph.D. thesis, Aston University.

DAYAN, P., & ZEMEL, R. S. 1995. Competition and multiple cause models. *Neural Computation*, **7**, 565–579.

DE BIE, T., & DE MOOR, B. 2002. On Two Classes of Alternatives to Canonical Correlation Analysis, using Mutual Information and Oblique Projections. *Proceedings of the 23rd Symposium on Information Theory in the Benelux, ITB'02*.

DEMPSTER, A. P., LAIRD, N. M., & RUBIN, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, **39**(1), 1–38.

ESCOBAR, M. D. 1994. Estimating normal means with a Dirichlet process prior. *Journal of the American Statistical Association*, **89**(425), 268–277.

FERGUSON, T. S. 1973. A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, **2**, 209–230.

FÖLDIÁK, P. 1990. Forming sparse representations by local anti-Hebbian learning. *Pages 165–170 of: Biological Cybernetics*, vol. 64.

FREY, B. 1998. *Graphical Models for Machine Learning and Digital Communication*. MIT Press.

FREY, B. J., P.DAYAN, & HINTON, G. E. 1997. A simple algorithm that discovers efficient perceptual codes. *In:* JENKIN, M., & HARRIS, I. R. (eds), *Computational and Psychophysical mechanisms of Visual Coding*. Cambridge University Press.

FRIEDMAN, N., MOSENZON, O., SLONIM, N., & TISHBY, N. 2001. Multivariate Information Bottleneck. *Pages 152–161 of:* HAYKIN, S. (ed), *Uncertainty in Artificial Intelligence: Proceedings of the Seventeenth Conference, UAI '01*. Morgan Kaufmann Publishers.

FYFE, C., & LEEN, G. 2006. Stochastic Processes for Canonical Correlation Analysis. *Pages 245–50 of: Proceedings of the 14th European Symposium of Artificial Neural Networks (ESANN)*.

GHAHRAMANI, Z., & BEAL, M. J. 2000. Variational inference for Bayesian mixtures of factor analyzers. *Pages 449–455 of:* SOLLER, S. A., LEEN, T.K., & MÜLLER, K. (eds), *Advances in Neural Information Processing Systems*, vol. 12. MIT Press.

GILKS, W. R., & WILD, P. 1992. Adaptive rejection sampling for Gibbs sampling. *Pages 337–348 of: Applied Statistics*, vol. 41.

GROCHOW, K., HERTZMANN, S.L. MARTIN A., & POPOVIC, Z. 2004. Style-based inverse kinematics. *ACM Trans. Graphics*, **23**(3), 522–531.

HAYKIN, S. 1994. *Neural Networks: A Comprehensive Foundation*. New York: Macmillan.

HINTON, G. E., REVOW, M., & DAYAN, P. 1995. Recognizing handwritten digits using mixtures of linear models. *Pages 1015 – 1022 of:* TESAURO, G., TOURETZKY, D. S., & LEEN, T. K. (eds), *Advances in Neural Information Processing Systems*, vol. 7. MIT Press.

HOTELLING, H. 1936. Relations between two sets of variates. *Biometrika*, **28**, 312–377.

JEBARA, T. 2001. *Discriminative, generative and imitative learning*. Ph.D. thesis, Media Lab, Massachusetts Institute of Technology.

JOLIFFE, I. T. 1986. *Principal Component Analysis*. New York: Springer-Verlag.

JORDAN, M. I. (ed). 1999. *Learning in Graphical Models*. MIT Press.

KAMBHATLA, N., & LEEN, T. K. 1997. Dimension reduction by local principal component analysis. *Neural Computation*, **9**(7), 1493–1516.

KAY, J. 1992. Feature discovery under contextual supervision using mutual information. *International Joint Conference on Neural Networks*, **4**, 79–84.

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

KLAMI, A., & KASKI, S. 2006. Generative models that discover dependencies between two data sets. *Pages 123–128 of:* MCLOONE, S., ADALI, T., LARSEN, J., HULLE, M. VAN, ROGERS, A., & DOUGLAS, S.C. (eds), *Machine Learning for Signal Processing XVI*. IEEE.

LAI, P. L., & FYFE, C. 1999. A neural implementation of canonical correlation analysis. *Neural Networks*, **12**, 1391–1397.

LAI, P. L., & FYFE, C. 2000. Kernel and Nonlinear Canonical Correlation Analysis. *International Journal of Neural Systems*, **10**(5), 365–377.

LAWRENCE, N. D. 2004. Gaussian Process Latent Variable Models for Visualization of High Dimensional Data. *Pages 329–336 of:* THRUN, S., SAUL, L., & SCHÖLKOPF, B. (eds), *Advances in Neural Information Processing Systems*, vol. 16. MIT Press.

LAWRENCE, N. D. 2005. Probabilistic Non-linear Principal Component Analysis with Gaussian Process Latent Variable Models. *Journal of Machine Learning Research*, **6**, 1783–1816.

LAWRENCE, N. D., SEEGER, M., & HERBICH, R. 2003. Fast sparse Gaussian process methods: The informative vector machine. *Pages 625–632 of:* BECKER, S., THRUN, S., & OBERMAYER, K. (eds), *Advances in Neural Information Processing Systems*, vol. 15. MIT Press.

LEEN, G., & FYFE, C. 2004a. Agent Wars with Artificial Immune Systems. *Pages 420 – 428 of:* RAUTERBERG, MATTHIAS (ed), *3rd International Conference on Entertainment Computing, ICEC2004*. Lecture Notes in Computer Science, vol. 3166. Springer-Berlin.

LEEN, G., & FYFE, C. 2004b. An investigation of alternative planning algorithms: Genetic algorithms, artificial immune systems and ant colony optimisation. *Pages 278–281 of: Conference on Computer Games: Design, AI and Education, CGAIDE2004*.

LEEN, G., & FYFE, C. 2005. Training an AI player to play pong using the GTM. *Pages 270 – 276 of: IEEE Symposium on Computational Intelligence and Games*.

LEEN, G., & FYFE, C. 2006. A Gaussian Process Latent Variable Model Formulation of Canonical Correlation Analysis. *Pages 413–418 of: Proceedings of the 14th European Symposium of Artificial Neural Networks (ESANN)*.

LINSKER, R. 1988. An Application of the Principle of Maximum Information Preservation to Linear Systems. *Pages 186–194 of:* TOURETZKY, D. (ed), *Advances in Neural Information Processing Systems*, vol. 1. Morgan Kaufmann.

MACKAY, D. J. C. 1995. Probable networks and plausiable predictions - a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, **6**(3), 469–505.

MACKAY, D. J. C. 1998. Introduction to Gaussian Processes. *Pages 133–166 of:* BISHOP, C. M. (ed), *Neural Networks and Machine Learning*. Springer-Verlag, Berlin.

MACKAY, D. J. C. 2003. *Information theory, Inference, and Learning Algorithms*. Cambridge University Press.

MARDIA, K. V., KENT, J. D., & BIBBY, J. M. 1979. *Multivariate Analysis*. Academic Press.

NEAL, R. M. 1991. *Bayesian mixture modeling by Monte Carlo simulation*. Tech. rept. CRG-TR-91-2. Department of Computer Science, University of Toronto.

NEAL, R. M. 1998. Assessing relevance determination methods using DELVE. *Pages 97–129 of:* BISHOP, C. M. (ed), *Neural Networks and Machine Learning*. Springer-Verlag.

O'HAGAN, A. 1978. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society, Series B*, **40**(1), 1–42.

PRINCIPE, J. C., XU, D., & III, J. W. FISHER. 2000. Information Theoretic Learning. *Chap. 7 of:* HAYKIN, S. (ed), *Unsupervised Adaptive Filtering*, vol. 1. John Wiley and Sons, New York.

RASMUSSEN, C. E., & WILLIAMS, C. K. I. 2006. *Gaussian Processes for Machine Learning*. MIT Press.

RASMUSSEN, C.E. 2000. The Infinite Gaussian Mixture Model. *Pages 554–560 of:* SOLLA, S. A., LEEN, T. K., & MÜLLER, K-R. (eds), *Advances in Neural Information Processing Systems*, vol. 12. MIT Press.

ROWEIS, S. T., SAUL, L. K., & HINTON, G. E. 2002. Global Coordination of Local Linear Models. *Pages 889–896 of:* DIETTERICH, T. G., BECKER, S., & GHAHRAMANI, Z. (eds), *Advances in Neural Information Processing Systems*, vol. 14. MIT Press.

SCHÖLKOPF, B., & SMOLA, A. J. 2002. *Learning with Kernels*. MIT Press.

SCHÖLKOPF, B., SMOLA, A., & MÜLLER, K.-R. 1998. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, **10**, 1299–1319.

SCHÖLKOPF, B., MIKA, S., BURGES, C., KNIRSCH, P., MÜLLER, K.-R., RATSCH, G., & SMOLA, A. J. 1999. Input space vs feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, **10**, 1000–1017.

SETHURAMAN, J. 1994. A Constructive Definition of Dirichlet Priors. *Pages 639–650 of: Statistica Sinica*, vol. 4.

SHANNON, C. 1948. A mathematical theory of communication. *Pages 379–423 of: Bell Systems Technical Journal*, vol. 27.

SHON, A. P., GROCHOW, K., HERTZMANN, A., & RAO, R. P. N. 2006. Learning shared latent structure for image synthesis and robotic imitation. *Pages 1233 – 1240 of:* WEISS, Y., SCHÖLKOPF, B., & PLATT, J. (eds), *Advances in Neural Information Processing Systems*, vol. 18. MIT Press.

SMOLA, A. J., MANGASARIAN, O. L., & SCHÖLKOPF, B. 1999. *Sparse Kernel Feature Analysis*. Tech. rept. 99-04. University of Wisconsin, Madison.

SMOLA, A. J., MIKA, S., SCHÖLKOPF, B., & WILLIAMSON, R. C. 2001. Regularized Principal Manifolds. *Journal of Machine Learning Research*, **1**, 179–209.

SNELSON, E., RASMUSSEN, C. E., & GHAHRAMANI, Z. 2004. Warped Gaussian Processes. *Pages 337–344 of:* THRUN, S., SAUL, L. K., & SCHÖLKOPF, B. (eds), *Advances in Neural Information Processing Systems*, vol. 16. MIT Press.

STUDHOLME, C., HAWKES, D.J., & HILL, D.L.G. 1999. An overlap invariant entropy measure of 3D medical image alignment. *Pattern Recognition*, **32**, 71–86.

SVENSÉN, M. 1998. *GTM: The Generative Topographic Mapping*. Ph.D. thesis, Aston University.

TEH, Y. W., SEEGER, M., & JORDAN, M. I. 2005. Semiparametric latent factor models. *Pages 333–340 of:* COWELL, ROBERT G., & GHAHRAMANI, ZOUBIN (eds), *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*. Society for Artificial Intelligence and Statistics.

TIPPING, M., & BISHOP, C. 1997. *Mixtures of Probabilistic Principal Component Analysers*. Tech. rept. NCRG/97/003. Neural Computing Research Group, Aston University.

TIPPING, M., & BISHOP, C. 1999. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, **21**(3), 611–622.

TISHBY, N., F.C, PEREIRA, & BIALEK, W. 1999. The Information Bottleneck method. *Pages 368–377 of:* HAJEK, B., & SREENIVAS, R. S. (eds), *Proc. of the 37th Annual Allerton Conference on Communication, Control and Computing*.

TORKKOLA, K. 2003. Feature Extraction by Non-parametric Mutual Information Maximization. *Journal of Machine Learning Research*, **3**, 1415–1438.

VAPNIK, V. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag.

VERBEEK, J., ROWEIS, S., & VLASSIS, N. 2004. Nonlinear CCA and PCA by alignment of local models. *Pages 297–304 of:* THRUN, S., SAUL, L. K., & SCHÖLKOPF, B. (eds), *Advances in Neural Information Processing Systems*, vol. 16.

VINOKOUROV, A., SHAWE-TAYLOR, J., & CRISTIANINI, N. 2003. Inferring a semantic representation of text via cross-language correlation analysis. *Pages 1473–1480 of:* BECKER, S., THRUN, S., & OBERMAYER, K. (eds), *Advances in Neural Information Processing Systems*, vol. 15. MIT Press.

VIOLA, P. A. 1995. *Alignment by Maximization of Mutual Information*. Ph.D. thesis, AI Lab, Massachusetts Institute of Technology.

WANG, C. 2007. Variational Bayesian Approach to Canonical Correlation Analysis. *IEEE Transactions on Neural Networks*, **18**(3), 905–910.

WANG, J., FLEET, D., & HERTZMANN, A. 2006. Gaussian Process Dynamical Models. *Pages 1441 – 1448 of:* WEISS, Y., SCHÖLKOPF, B., & PLATT, J. (eds), *Advances in Neural Information Processing Systems*, vol. 18. MIT Press.

WANG, J. M-C. 2005. *Gaussian Process Dynamical Models for Human Motion*. M.Phil. thesis, Graduate Department of Computer Science, University of Toronto.

WEST, M., MÜLLER, P., & ESCOBAR, M. D. 1994. Hierarchical priors and mixture models with applications in regression and density estimation. *Pages 363–386 of:* FREEMAN, P. R., & SMITH, A. F. M (eds), *Aspects of Uncertainty*.

WILLIAMS, C. K. I., & RASMUSSEN, C. E. 1996. Gaussian processes for regression . *Pages 514–520 of:* TOURETZKY, D., MOZER, M., & HASSELMO, M. (eds), *Advances in Neural Information Processing Systems*, vol. 8. MIT Press.