Learning with structured covariance matrices in linear Gaussian models

Alfredo Kalaitzis

Department of Computer Science

University of Sheffield

A thesis submitted for the degree of

Doctor of Philosophy

February 2013

Abstract

We study structured covariance matrices in a Gaussian setting for a variety of data analysis scenarios. Despite its simplistic nature, we argue for the broad applicability of the Gaussian family through its second order statistics. We focus on three types of common structures in the machine learning literature: covariance *functions*, *low-rank* and *sparse inverses* covariances. Our contributions boil down to combining these structures and designing algorithms for maximum-likelihood or MAP fitting: for instance, we use covariance functions in Gaussian processes to encode the temporal structure in a gene-expression time-series, with any residual structure generating iid noise. More generally, for a low-rank residual structure (correlated residuals) we introduce the *residual component analysis* framework: based on a generalised eigenvalue problem, it decomposes the residual low-rank term given a partial explanation of the covariance. In this example the explained covariance would be an RBF kernel, but it can be any positive-definite matrix. Another example is the *low-rank plus sparse-inverse* composition for structure learning of GMRFs in the presence of confounding latent variables. We also study RCA as a novel link between classical low-rank methods and modern probabilistic counterparts: the geometry of oblique projections shows how PCA, CCA and *linear discriminant analysis* reduce to RCA. Also inter-battery factor analysis, a precursor of multi-view learning, is reduced to an iterative application of RCA. Finally, we touch on structured precisions of matrix-normal models based on the Cartesian factorisation of graphs, with appealing properties for regression problems and interpretability. In all cases, experimental results and simulations demonstrate the performance of the different methods proposed.

Learning with structured covariance matrices in linear Gaussian models

Alfredo Kalaitzis

Department of Computer Science

University of Sheffield

A thesis submitted for the degree of

Doctor of Philosophy

February 2013

Declaration of Authorship

I, Alfredo Kalaitzis, declare that this thesis titled, “Learning with structured covariance matrices in linear Gaussian models” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Acknowledgements

I would like to thank my supervisor, Prof. Neil Lawrence, for his patient and encouraging guidance, his advice on all aspects of academic life and his constant stream of ideas while allowing me my own mode of work. For all the effort he put in guaranteeing us, his students, the smoothest of transitions during his promotion and our move to Sheffield.

I would like to thank Michalis Titsias, Kevin Sharp and Mauricio Alvarez for helpful discussions and technical guidance during my one year in Manchester. The same thanks go to Magnus Rattray, Marta Milo, James Hensman, Nicholas Durrande, Nicolo Fusi and Jaakko Peltonen during my other two years in Sheffield. I also want to thank Diego di Bernardo for his useful feedback on the experimental data of Chapter 2 and John Lafferty for the technical material that sparked the ideas in Chapter 6.

I had the pleasure of working in two wonderful environments thanks to the staff and head of the MLO group, Prof. Jonathan Shapiro, as well as the staff and director of SITraN, Prof. Pamela Shaw.

For reasons I don't have to mention, I thank colleagues and friends from both labs (alphabetically): Ricardo Andrade, Richard Allmendinger, Mauricio Alvarez, Arjun Chandra, Andreas Damianou, Nicholas Durrande, Nicolo Fusi, Peter Glaus, James Hensman, Antti Honkela, Ciira Maina, Jens Nielsen, Jaakko Peltonen, Adam Pocock, Arif Rahman, Jon Roberts, Kevin Sharp, Manuela Zanda and not least, Michalis Titsias.

Finally, I would like to thank the Engineering and Physical Sciences

Research Council for providing the funding which allowed me to undertake this research, and together with the PASCAL Network of Excellence enabled me to attend conferences, workshops and summer schools.

I dedicate this thesis to my grandfather and my late grandmother for giving me the opportunity to pursue my dream. To Laura, for proofreading my meaningless drafts and sitting through my talk rehearsals. I cherish your never-ending support, encouragement and patience. To my brother Kostis, for choosing to move his life to Sheffield and, in the process, being family in proximity; a rare privilege for any foreign student.

Abstract

We study structured covariance matrices in a Gaussian setting for a variety of data analysis scenarios. Despite its simplistic nature, we argue for the broad applicability of the Gaussian family through its second order statistics. We focus on three types of common structures in the machine learning literature: covariance *functions*, *low-rank* and *sparse inverses* covariances. Our contributions boil down to combining these structures and designing algorithms for maximum-likelihood or MAP fitting: for instance, we use covariance functions in Gaussian processes to encode the temporal structure in a gene-expression time-series, with any residual structure generating iid noise. More generally, for a low-rank residual structure (correlated residuals) we introduce the *residual component analysis* framework: based on a generalised eigenvalue problem, it decomposes the residual low-rank term given a partial explanation of the covariance. In this example the explained covariance would be an RBF kernel, but it can be any positive-definite matrix. Another example is the *low-rank plus sparse-inverse* composition for structure learning of GMRFs in the presence of confounding latent variables. We also study RCA as a novel link between classical low-rank methods and modern probabilistic counterparts: the geometry of oblique projections shows how PCA, CCA and *linear discriminant analysis* reduce to RCA. Also inter-battery factor analysis, a precursor of multi-view learning, is reduced to an iterative application of RCA. Finally, we touch on structured precisions of matrix-normal models based on the Cartesian factorisation of graphs, with appealing properties for regression problems and interpretability. In all cases, experimental results and simulations demonstrate the performance of the different methods proposed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Gaussian family of distributions seems simplistic with only two moments to fit, though with easily interpretable parameters. Its tails decay too rapidly for many real-world distributions. In the spirit of George Box's most famous quote[1], we will argue through examples that despite its simplicity and unrealism, the standard multivariate Gaussian approach remains ever-useful to statisticians and machine learners.

As the title hints, we will focus on the *interactions* between whatever covariates (or interchangeably, features, observables, independent variables, inputs) a data analysis is concerned with. Precisely, *structured* second-order statistics as encoded by the multivariate Gaussian family will be our toolbox. For a p-dimensional Gaussian, its $p \times p$ covariance matrix is very expressive and interpretable: a zero in the covariance implies a *marginal* independence constraint, a zero in the inverse-covariance implies a *conditional* independence constraint and its spectral properties can shed light to *hidden* variables. By "structured" we mean any restrictions that we will impose on the functional form of the covariance matrix. We will restrict our attention to *linear* relationships between any Gaussian random variables, so all of the models presented will be *linear-Gaussian*, see also a seminal work by Roweis & Ghahramani [1999]. Approaches for converting the linear to non-linear (for instance, generalised linear models,

[1] "All models are wrong but some are useful."

GPLVM, etc) or the inverse (for instance, Gaussian copulas, etc) are potential directions to eventually take. In each chapter we will motivate a particular structure type or combinations of such, namely: covariance functions, low-rank and/or sparse-inverse covariances, Kronecker-products or Kronecker-sums of precision (inverse-covariance) matrices. To these ends, with the exception of Gaussian processes (GP), we will propose novel — and in some cases re-discovered — ways to point-estimate (via maximum likelihood or a posteriori) these structures. Fully Bayesian extensions would also be of interest to the community.

Covariance functions The Gaussian process is one of the success stories of the Gaussian family in machine learning [Rasmussen & Williams, 2006]. Covariance functions or kernels are now a mainstream line of research partly due to the expansive application of Gaussian processes, first in the geostatistics community (non-linear regression under the name of *kriging*) and later in machine learning for regression and classification. Therefore in Chapter 2 we kick off the story with a regression problem that we faced early on, under the guise of ranking genes based on the differential expression of their time-series (or microarray) data. The regression problem was solved by modeling the series with a Gaussian distribution whose covariance is parametrised by a "vanilla" RBF[1] kernel. But the true contribution of the chapter is perhaps the ranking of genes based on ratios of GP marginal likelihoods. The chapter is a re-edited version of [Kalaitzis & Lawrence, 2011b] and besides presenting the methodology it is also a self-contained introduction to GP regression.

Low rank Perhaps the protagonist of this thesis and the oldest in origin within the chosen topics is the low-rank type of covariance structure. To conceptually tie it with the regression problem above, consider the following problem of correlation effects in the expression series – for instance, due to a normalisation – that might be hidden in the microarray data. For each gene, the GP either explains the empirical gene-expression variance with the smooth RBF covariance

[1] A.k.a. the *double-exponential*, *Gaussian* or as Neil Lawrence prefers, the *exponentiated quadratic* kernel.

function, or simply as independent Gaussian spherical noise. It is reasonable to assume that some structure in the measurement error (that is, a colored noise effect) remains to be recovered. This would amount to some additive *low-rank* structure in the covariance of the time-points. This motivates our *residual component analysis* (RCA) framework in Chapter 3: the recovery of lower-dimensional components that explain the structured noise effect hidden in the data, given a partial explanation (fixed additive part) of the marginal covariance.

Chapter 4 will deal with additional theoretical aspects of RCA and in particular its role as a novel link between classical (non-probabilistic) low-rank methods and their modern probabilistic counterparts. For instance, we will show how RCA and its probabilistic variant generalise classical/probabilistic PCA/CCA respectively. Linear discriminant analysis will also reduce to RCA, thus strengthening the marriage of unsupervised and supervised learning. Inspired by the signal processing literature, the geometry of *oblique projections* will bind them all. RCA will also be used to extend, as opposed to generalise. For instance, we will show how an iterative application of RCA (*iterative-RCA*) can be used to analyse disjoint set of covariates with paired samples that supposedly measure overlapping sets of latent factors, a problem known in machine learning as *multi-view learning*. In this way, we re-invented a re-invention: our iterative-RCA is equivalent to the *extended-CCA* model of Klami & Kaski [2006] which is equivalent of the *inter-battery factor analysis* model studied in the statistics literature [Tucker, 1958].

We will discuss a few applications of RCA in Chapter 5. Among them is a simple demonstration that will tie RCA with the use of GPs in Chapter 2: explaining away the trained covariance of a GP (defined by a RBF) on concatenated time-series from two separate experiments. The residual structure will serve as the basis for measuring the differential expression across the experiments. Perhaps the most promising application is the reconstruction of regulatory networks of genes from protein-signaling data. Usually such data are confounded by low-rank effects: structured (that is, correlated) measurement deviations introduced by measuring under heterogeneous experimental conditions (for instance, due to different cell perturbations or platforms or even labs). RCA will help to reduce the low-rank effects and ultimately recover more accurately the sparse conditional dependency structure (equivalently, the sparse precision matrix of the joint Gaussian). But the methodology to actually recover this Markov network will bring us to the next subject of covariance structures.

Sparse inverse-covariances In the later half of this thesis (Chapters 5, 6) we will depart from the purely *directed* nature of such generative graphical models induced by low-rank structures, and concentrate on learning the structures of purely *undirected* graphs or *Markov networks*. This problem is effectively solved for the purposes of point-estimation, assuming that the true distribution is Markov with respect to a Gaussian graphical model [Banerjee *et al.*, 2008; Friedman *et al.*, 2008]. The challenge presents itself in the form of *unknown latent* factors in the graph (by "unknown" we mean that one simply postulates their existence); one which we will try to tackle with a *low-rank plus sparse-inverse* covariance structure. We will present no contribution with regards to sparse-inverse selection or lasso optimisation per se, as this theory as firmly founded for our purposes [Tibshirani, 1996]. Chapters 3,4 and 5 are collectively an extended version of [Kalaitzis & Lawrence, 2011a, 2012].

Kronecker-products and Kronecker-sums of sparse inverses-covariances In the 6th and final chapter we will describe some work in progress on purely sparse-inverse structures in matrix-Gaussians (or matrix-normals), that is, Gaussian distributions over random matrices. Their potentially large covariances can exploit ideas from algebraic graph theory giving modeling and algorithmic benefits for learning simultaneously two sparse conditional dependency structures, one over the rows of a matrix-sample and one over its columns. Finally, in chapter 7 we will close with a summary of the main ideas and results of this thesis and outline some ideas for future research.

Chapter 2

Temporal covariance structures for ranking differential expression

The analysis of gene expression from time series underpins many biological studies. Two basic forms of analysis recur for data of this type: removing inactive (quiet) genes from the study and determining which genes are differentially expressed. Here, the problem is one of ranking genes based on the differential expression of their time-series (or microarray) data. Often these analysis stages are applied disregarding the fact that the data is drawn from a time series. In this chapter we propose a temporal covariance structure to account for the underlying temporal nature of the data based on a Gaussian process (GP) [Kalaitzis & Lawrence, 2011b].

Results We review GP regression, for estimating the continuous trajectories underlying the gene expression time-series, in section 2.2. We present a simple approach which can be used to filter quiet genes, or for the case of time series in the form of expression ratios, quantify differential expression. We assess the rankings produced by our regression framework through ROC curves and compare them to a recently proposed hierarchical Bayesian model (BATS) in section 2.3.

We compare on both simulated and experimental data showing that the proposed approach considerably outperforms the current state of the art.

Conclusions We argue in section 2.4 that GPs offer an attractive trade-off between efficiency and usability for the analysis of microarray time series. The GP framework offers a natural way of handling biological replicates and missing values and provides confidence intervals along the estimated curves of gene expression. Therefore, we believe that GPs should be a standard tool in the analysis of gene expression time series.

2.1 Background

Gene expression profiles give a snapshot of mRNA concentration levels as encoded by the genes of an organism under given experimental conditions. Early studies of this data often focused on a single point in time which biologists assumed to be critical along the gene regulation process after the perturbation. However, the *static* nature of such experiments severely restricts the inferences that can be made about the underlying dynamical system.

With the decreasing cost of gene expression microarrays time series experiments have become commonplace giving a far broader picture of the gene regulation process. Such time series are often irregularly sampled and may involve differing numbers of replicates at each time point [Lönnstedt & Speed, 2002]. The experimental conditions under which gene expression measurements are taken cannot be perfectly controlled leading the signals of interest to be corrupted by noise, either of biological origin or arising through the measurement process.

Primary analysis of gene expression profiles is often dominated by methods targeted at *static* experiments, i.e. gene expression measured on a single time-point, that treat time as an additional experimental factor [Dudoit *et al.*, 2002; Efron *et al.*, 2001; Friedman *et al.*, 2000; Kerr *et al.*, 2000; Lönnstedt & Speed, 2002; Spellman *et al.*, 1998]. However, where possible, it would seem sensible to consider methods that can account for the special nature of time course data.

Such methods can take advantage of the particular statistical constraints that are imposed on data that is naturally ordered [Angelini *et al.*, 2007, 2008; Bar-Joseph *et al.*, 2003; Ernst *et al.*, 2005; Storey *et al.*, 2005; Tai & Speed, 2006].

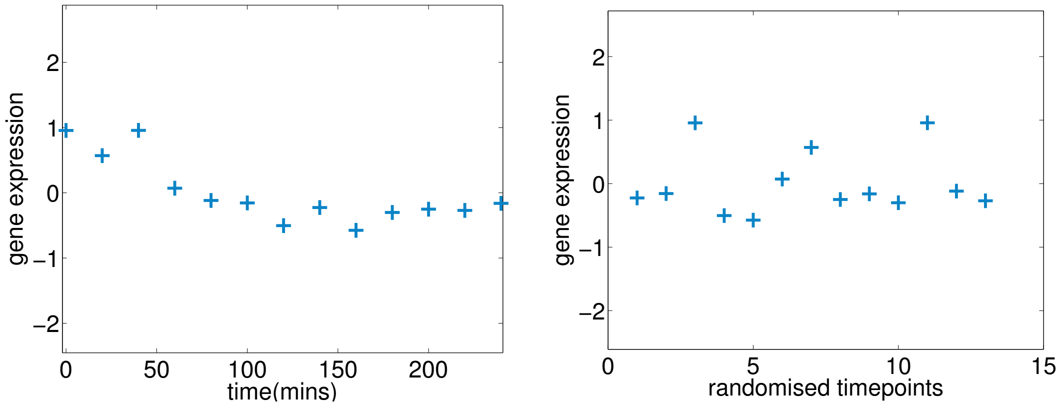
The analysis of gene expression microarray time-series has been a stepping stone to important problems in systems biology such as the genome-wide identification of direct targets of transcription factors [Della Gatta *et al.*, 2008; Honkela *et al.*, 2010] and the full reconstruction of gene regulatory networks [Bansal *et al.*, 2006; Finkenstadt *et al.*, 2008]. A more comprehensive review on the motivations and methods of analysis of time-course gene expression data can be found in [Bar-Joseph, 2004].

2.1.1 Testing for expression

A primary stage of analysis is to characterize the activity of each gene in an experiment. Removing inactive or *quiet* genes (genes which show negligible changes in mRNA concentration levels in response to treatments/perturbations) allows the focus to dwell on genes that have responded to treatment. We can consider two experimental set ups. Firstly, we may be attempting to measure the absolute level of gene expression (for example using Affymetrix GeneChip microarrays). In this case a quiet gene would be one whose expression level is indistinguishable from noise. Alternatively, we might be may be hybridizing two samples to the same array and quantifying the ratio of the expression levels. Here a quiet gene would be one which is showing a similar response in both hybridized samples. In either case we consider such expression profiles will consist principally of *noise*. Removing such genes will often have benign effects later in the processing pipeline. However, mistaken removal of profiles can clearly compromise any further downstream analysis. If the temporal nature of the data is ignored, our ability to detect such phenomena can be severely compromised. An example can be seen in Figure 2.1, where the temporal information is removed from an experimental profile by randomly reordering its expression samples. Disregarding the temporal correlation between measurements, hinders our ability to assess the profile due to critical inherent traits of the signal being lost such as the speed and scale of variation.

Failure to capture the signal in a profile, irrespective of the amount of embedded noise, may be partially due to *temporal aggregation* effects, meaning that the coarse sampling of gene expression or the sampling rates do not match the natural rates of change in mRNA concentrations [Bay *et al.*, 2004]. For these reasons, the classification scheme of differential expression in this paper is focused on reaching a high *true positive rate* (TPR, *sensitivity* or *recall*) and is to serve as a pre-processing tool prior to more involved analysis of time-course microarray data. In this work we distinguish between *two-sample* testing and experiments where *control* and *treated* cases are directly-hybridized on the microarray (For brevity, we shall refer to experiments with such setups as *one-sample testing*). The *two-sample* setup is a common experimental setup in which two groups of sample replicates are used [Della Gatta *et al.*, 2008; Stegle *et al.*, 2010]; one being under the treatment effect of interest and the other being the control group, so to recover the most active genes under a treatment one may be interested in testing for the statistical significance of a treated profile being differentially expressed with respect to its control counterpart. Other studies use data from a *one-sample* setup [Angelini *et al.*, 2007, 2008], in which the *control* and *treated* cases are directly hybridized on a microarray and the measurements are normalized log fold-changes between the two output channels of the microarray [Schena *et al.*, 1995], so the analogous goal is to test for the statistical significance of having a non-zero signal.

A recent significant contribution in estimating and ranking the differential expression of time-series in a *one-sample* setup is the hierarchical Bayesian model for the analysis of gene expression time-series (BATS) [Angelini *et al.*, 2007, 2008]. The framework offers fast computations through exact computations of Bayesian inference, to the cost of making a considerable number of biological assumptions, see section 2.3.2.



(a)

(b)

Figure 2.1: Temporal information removed from the profile of gene Cyp1b1 in the experimental mouse data. **(a)** The centred profile of the gene *Cyp1b1* (probeID 1416612_at in the *GSE10562* dataset). The blue crosses represent zero-mean hybridised gene expression in time of measurement (log2 ratios between treatment and control). **(b)** The same profile with its timepoints randomly shuffled.

2.1.2 Gene expression analysis with Gaussian processes

Gaussian processes (GP) [MacKay, 2003; Rasmussen & Williams, 2006] offer an easy to implement approach for quantifying the true signal and noise embedded in a gene expression time-series, and thus allow us to rank the differential expression of the gene profile. We initially motivated GPs as Gaussians with a particular (temporal) type of covariance structure over the expression time-points. More generally in the context of the Gaussian family of distributions, a Gaussian process is the natural generalisation of a multivariate Gaussian distribution to a Gaussian distribution over a space of a *specific family of functions* — a family defined by a *covariance function* or *kernel*, that is, a similarity metric between datapoints. Roughly speaking, viewing a function as an infinite-dimensional vector, allows one to represent that function as a point in an infinite-dimensional *space of a specific class of functions*, and a Gaussian process as an infinite-dimensional Gaussian distribution over that space.

In the context of expression trajectory estimation, a Gaussian process coupled with the *squared-exponential* covariance function (or *radial basis function*, RBF)

— a standard covariance function used in regression tasks — makes the reasonable assumption that the underlying true signal in a profile is a *smooth* function [Yuan, 2006], that is, an infinitely differentiable function. This property endows the GP with great flexibility in capturing the underlying signals, without imposing strong modeling assumptions (as in, a finite number of basis functions in BATS) but may also erroneously pick up spurious patterns (false positives) should the time-course profiles suffer from temporal aggregation effects. From a generative viewpoint, the profiles are assumed to be corrupted with additive independent spherical (iid) Gaussian noise. This property makes the GP an attractive tool for bootstrapping simulated biological replicates [Kirk & Stumpf, 2009].

In a different context, Gaussian process priors have been used for modeling transcriptional regulation. For example, Lawrence *et al.* [2007], while using the time-course expression of a-priori known direct targets (genes) of a transcription-factor, the authors went one step further and inferred the concentration rates of the transcription-factor protein itself and Gao *et al.* [2008] extended the same model for the case of regulatory repression.

The ever-lingering issue of outliers in time series is still critical, but is not addressed here as there is significant literature on this issue, in the context of GP regression, complementary to this work. For instance, Stegle *et al.* [2009, 2010] proposed a probabilistic model using Gaussian processes with a robust noise model specialised for two-sample testing to detect *intervals* of differential expression, whereas the present work optionally focuses on *one-sample* testing, to rank the differential expression and ultimately detect *quiet/active* genes. Other examples can also be easily applied; Tipping & Lawrence [2005] used a Student-t distribution as the robust noise model in the regression framework along with variational approximations to make the inference mechanism tractable, and Vanhatalo *et al.* [2009] used a Student-t observation model with Laplace approximations for inference.

In this case study, the standard GP regression framework is straightforward to use, with a minimal need for manual tweaking of a few hyper-parameters. Section 2.2 describes the GP regression framework in detail.

2.2 Methodology

The modeling of time-course microarray data with GPs is not a new idea (see section 2.1, **Background**). In this section we review the methodology for estimating the continuous trajectory of a gene expression through GP regression. This is followed by a likelihood-ratio approach to ranking the differential expression of a gene, in section 2.2.2. The following section contains some key GP theory, borrowing from chapters 45 and 2 in [MacKay, 2003; Rasmussen & Williams, 2006, respectively].

2.2.1 The Gaussian process model

The main idea is to treat trajectory estimation, given some noisy output observations (gene expression), as an interpolation problem on functions of one (time) dimension. By assuming that the observations are jointly Gaussian-distributed with Gaussian iid noise, the computations for prediction become tractable and involve only the manipulation of linear algebra rules.

A finite parametric model

We gradually introduce the GP regression model, starting from a linear regression model with inputs $\mathbf{x} \in \mathbb{R}^p$ mapped to some feature space defined by $\boldsymbol{\phi} = \phi(\mathbf{x})$:

$$f(\mathbf{x}) = \boldsymbol{\phi}^\top \mathbf{w}, \qquad y = f(\mathbf{x}) + \epsilon \ . \tag{2.1}$$

For example, $\phi(x) = (1,\, x,\, x^2\,)^\top$ maps a line to a quadratic curve. In our case, gene expression is measured at timepoints $\{x_i\}_{1..n}$, to form a profile of observations $\{y_i\}_{1..n}$. The input and output dimensionalities are one. The (time) inputs are mapped to features $\{\boldsymbol{\phi}(x_i)\}_{1..n}$. We assume that the observations are contaminated with Gaussian iid noise of zero mean and variance σ^2:

$$\epsilon \sim \mathcal{N}\left(0, \sigma^2\right) . \tag{2.2}$$

Then the likelihood of the observations $\mathbf{y} = \{y_i\}_{1..n}$, given inputs[1] $\mathbf{x} = \{x_i\}_{1..n}$ and parameters $\mathbf{w}$, is Gaussian:

$$\begin{aligned}
p(\mathbf{y} \,|\, \mathbf{x}, \mathbf{w}) &= \prod_{i=1}^{n} (2\pi\sigma^2)^{-1/2} \exp\left\{ -\frac{(y_i - \boldsymbol{\phi}_i^\top \mathbf{w})^2}{2\sigma^2} \right\} \\
&= (2\pi\sigma^2)^{-n/2} \exp\left\{ -\frac{1}{2\sigma^2} (\mathbf{y} - \boldsymbol{\Phi}\mathbf{w})^\top (\mathbf{y} - \boldsymbol{\Phi}\mathbf{w}) \right\} \\
&= \mathcal{N}(\mathbf{y} \,|\, \boldsymbol{\Phi}\, \mathbf{w}, \sigma^2 \mathbf{I}) \ .
\end{aligned}$$

Here, we assume that the observations are conditionally independent given the inputs, that is, the likelihood factorises across the y_i.

Bayesian linear regression

Now we wish to include some prior belief about the parameters $\mathbf{w}$, by specifying a zero mean spherical Gaussian as a *prior* distribution over the parameters:

$$\mathbf{w} \sim \mathcal{N}\left(0, \sigma_w^2\right) .$$

Now we can integrate out the parameters from the joint distribution $p(\mathbf{y}, \mathbf{w} \,|\, \mathbf{x})$ to get the *marginal* likelihood

$$p(\mathbf{y} \,|\, \mathbf{x}) = \int \mathrm{d}\mathbf{w} \ p(\mathbf{y} \,|\, \mathbf{x}, \mathbf{w}) p(\mathbf{w}) \ . \tag{2.3}$$

The marginal is also Gaussian with mean and covariance

$$\mathbb{E}\left[\mathbf{y} \,|\, \mathbf{x}\right] \quad = \quad \boldsymbol{\Phi}\mathbb{E}\left[\mathbf{w}\right] + \mathbb{E}\left[\boldsymbol{\epsilon}\right] \quad = \quad \mathbf{0} \tag{2.4}$$

$$\begin{aligned}
\mathrm{var}\left[\mathbf{y} \,|\, \mathbf{x}\right] \quad &= \quad \boldsymbol{\Phi} \ \mathrm{var}\left[\mathbf{w}\right] \boldsymbol{\Phi}^\top + \mathrm{var}\left[\epsilon\right] \\
&= \quad \sigma_w^2 \boldsymbol{\Phi}\boldsymbol{\Phi}^\top + \sigma^2 \mathbf{I} \qquad (2.5) \\
&\triangleq \quad \mathbf{K}_f + \sigma^2 \mathbf{I} \quad \triangleq \quad \mathbf{K}_y
\end{aligned}$$

[1] Normally, we use the notation $\mathbf{x}$ to denote a single datapoint of dimension p. In this section, we diverge temporarily from this notation to denote a *collection* of inputs.

$$\begin{aligned} p(\mathbf{y}\,|\,\mathbf{x}) &= \mathcal{N}(\mathbf{y}\,|\,\mathbf{0}, \mathbf{K}_y) \\ &= (2\pi)^{-n/2}|\mathbf{K}_y|^{-1/2}\exp\left\{-\frac{1}{2}\mathbf{y}^\top\mathbf{K}_y^{-1}\mathbf{y}\right\}, \end{aligned} \tag{2.6}$$

where $\mathbf{K}_y$ is defined as the covariance from eq. (2.5) and $\mathbf{K}_f$ as the noiseless signal covariance.

Why Bayesian? Recall that the structure of the covariance in eq. (2.5) relies on the choice of mapping ϕ. This can be a mapping to variable-order polynomials, by adjusting the polynomial degree *hyperparameter*, or a RBF (radial basis function) with a variable *lengthscale*. While these different classes of features give different classes of models, within one class we can *compare* or rank different models (different choices of hyperparameters) through the marginal likelihood in eq. (2.6). This is made possible by the marginalisation in eq. (2.3), which is a weighted average over the parameters $\mathbf{w}$, and the Gaussianity assumptions of the data likelihood and prior give the integral a closed form solution. At the same time, the Bayesian approach reduces *overfitting* on the data, without having to apply explicitly a *regulariser* to the data fit term. In fact, the marginal likelihood implicitly penalises overly complex models (e.g. high degree polynomials) as the prior assumes a lower probability density for such values of $\mathbf{w}$, see sections 2.8 and 5.4 in [MacKay, 2003; Rasmussen & Williams, 2006, respectively].

For $\mathbf{K}_y$ to be a valid covariance matrix of the GP, it must satisfy the following conditions:

- **Kolmogorov consistency**: satisfied when $K_{ij} = k(x_i, x_j)$ for some *covariance function* $k : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$, such that $\mathbf{K}$ is positive semidefinite — that is, $\mathbf{y}^\top\mathbf{K}\mathbf{y} \geq 0$ for any $\mathbf{y}$ or, equivalently, the eigenvalues of $\mathbf{K}$ are non-negative.
- **Exchangeability**: satisfied when the data are iid. This implies that the order in which the data become available has no impact on the marginal distribution.

Definition of the Gaussian process

More formally, *a Gaussian process is a collection of random variables (a stochastic process), such that the joint distribution over any finite subset,*

$$p\left(y_1, y_2, \dots, y_n\right),$$

is Gaussian and its covariance satisfies the Kolmogorov consistency.

If we remove the noise term $\sigma^2\mathbf{I}$ from $\mathbf{K}_y$ in eq. (2.5), we get noiseless predictions of $f(x)$ rather than $y(x)$, see eq. (2.1). However, when dealing with finite parameter spaces, $\mathbf{K}_f$ may be *ill-conditioned* (determinant close to zero), so the constant diagonal noise term increases the eigenvalues just enough to make $\mathbf{K}_y$ invertible.

Now we can view the GP as a Gaussian prior distribution over the function values $\mathbf{f}$ for inputs $\mathbf{x}$, by rewriting eq. (2.6):

$$p(\mathbf{f}\,|\,\mathbf{x}) = (2\pi)^{-n/2}|\mathbf{K}_f|^{-1/2}\left\{-\frac{1}{2}(\mathbf{f}-\mathbf{m})^{\top}\mathbf{K}_f^{-1}(\mathbf{f}-\mathbf{m})\right\}\ . \tag{2.7}$$

But more generally, it turns out that the GP can be safely defined as a prior distribution over functions f:

$$\begin{aligned} f(x) &\sim \mathcal{GP}\left(m(x),\ k(x,x')\right) \\ m(x) &= \mathbb{E}\left[f(x)\right] && (2.8) \\ k(x,x') &= \mathbb{E}\left[(f(x)-m(x))(f(x')-m(x'))\right]\ , && (2.9) \end{aligned}$$

where m is the *mean function* (usually defined as the zero function) and k is the *covariance function* satisfying the Kolmogorov consistency. Fortunately, in practice, we only have to handle a finite number of dimensions of the GP (eq. 2.7), as we can only access a finite collection of inputs $\mathbf{x}$.

2.2.2 The squared-exponential kernel

In this case study we use the univariate version of the SE (squared-exponential) or RBF kernel. Before embarking on its analysis, the reader should be aware of the existing wide variety of kernel families, and combinations of them. A comprehensive review of covariance functions can be found in [Rasmussen & Williams, 2006, chapter 4].

Derivation

In section 2.2.1 we mentioned the possibility of an ill-conditioned covariance matrix, in the case of a finite parametric model. We can see this from eq. (2.5), where $\mathbf{K}_f$ can have at most as many non-zero eigenvalues as the number of parameters in the model. Hence for any problem of any given size, the matrix is positive semidefinite. Ensuring that $\mathbf{K}_f$ is positive *definite*, involves adding the diagonal noise term to the covariance.

On the other hand, it can be shown that with a particular feature space (with an infinite number of RBFs), when the features are integrated out then the covariance between the datapoints is expressed by a covariance function instead of the features. To show this, first we consider a feature defined by the RBF ϕ_{x_c} centred at point x_c, such that $\boldsymbol{\phi}_{x_c} = (\phi_{x_c}(x_1), ..., \phi_{x_c}(x_n))^\top$ for $c \in \{1, ..., n\}$, so $\mathbf{\Phi} = (\boldsymbol{\phi}_{x_1}, ..., \boldsymbol{\phi}_{x_n})$. We express the covariance matrix $\mathbf{K}_f$ in terms of a decomposition of outer-products:

$$\mathbf{K}_f = \sigma_w^2 \mathbf{\Phi}\mathbf{\Phi}^\top = \sigma_w^2 \sum_c \boldsymbol{\phi}_{x_c} \boldsymbol{\phi}_{x_c}^\top \ ,$$

which is equivalent to

$$K_{f,ij} = k(x_i, x_j) = \sigma_w^2 \sum_c \phi_{x_c}(x_i)\phi_{x_c}(x_j) \ . \tag{2.10}$$

Here, the number of features (complexity) k is equal to the number of datapoints n. With an infinite number of centers (features) on the real line, the limit converges [MacKay, 1998]:

$$\begin{aligned} k(x_i, x_j) &= \lim_{k\to\infty} \frac{\sigma_w^2}{k} \sum_{c=1}^{k} \phi_{x_c}(x_i)\phi_{x_c}(x_j) \\ &= \sigma_w^2 \int_{-\infty}^{\infty} \mathrm{d}x_c \; \phi_{x_c}(x_i)\phi_{x_c}(x_j) \\ &= \sigma_w^2 \int_{-\infty}^{\infty} \mathrm{d}x_c \;\exp\left\{-\frac{(x_i - x_c)^2}{2r^2}\right\} \exp\left\{-\frac{(x_j - x_c)^2}{2r^2}\right\} \\ &= \sqrt{\pi r^2}\; \sigma_w^2 \exp\left\{-\frac{(x_i \;-\; x_j)^2}{4r^2}\right\} \\ &= \sigma_f^2 \exp\left\{-\frac{(x_i \;-\; x_j)^2}{2\ell^2}\right\} \;. \end{aligned} \tag{2.11}$$

Analysis

It turns out that by integrating out the feature centers, we end up with another scaled RBF function with either input as its center, that is, $k_{x_i}(x_j) = k_{x_j}(x_i) = k(x_i, x_j)$. This *covariance function or kernel* satisfies the Kolmogorov consistency and is known as the SE or RBF kernel. The factor $\sigma_f^2 \triangleq \sqrt{\pi r^2}\, \sigma_w^2$ acts as a *signal variance* and $\ell^2 \triangleq 2r^2$ as the *lengthscale* of this standard form of the univariate SE covariance function. In practice, we use the noisy version:

$$K_{y,ij} = \sigma_f^2 \exp\left\{-\frac{(x_i - x_j)^2}{2\ell^2}\right\} + \sigma^2\delta_{ij} \;, \tag{2.12}$$

where δ_{ij} is the Kronecker delta function which is one for $i = j$ and zero otherwise.

RKHS kernel As a side note, the RBF feature k_x is a member of a *reproducing kernel Hilbert space* (RKHS). A Hilbert space $\mathcal{H}$ is a space with an inner product $\langle ., . \rangle_{\mathcal{H}}$ and it is *complete*[1] with respect to the norm induced by this inner product[3].

[1] The space must contain the limits of all Cauchy[2] sequences of elements in $\mathcal{H}$.

[2] A sequence $x_1, x_2, ... \in \mathcal{H}$ is Cauchy if for an arbitrarily small $\epsilon > 0$, the distance $d(x_i, x_j)$ (as induced by the norm) always gets smaller than ϵ for some i onwards in the sequence.

[3] In our case, we have a space of real functions $f : \mathbb{R} \to \mathbb{R}$ with norm $||f||_{\mathcal{H}}^2 = \langle f, f\rangle_{\mathcal{H}}$.

It is an RKHS, if there is some function $k : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ such that:

- For all $x \in \mathbb{R}$, $k(x, x')$ as a function of x' is a member of $\mathcal{H}$.
- For all $x \in \mathbb{R}$, k_x is the evaluation representer; that is, for any $f \in \mathcal{H}$, $\langle f, k_x \rangle_{\mathcal{H}} = f(x)$. In this case we say that k is a *reproducing kernel.*

For more details on the properties of RKHS spaces, see [Berlinet & Thomas-Agnan, 2004; Rasmussen & Williams, 2006; Schölkopf & Smola, 2002; Wegman, 1988].

Kernel trick Since $k_{x_i}(.) = k(x_i, .)$ and $k_{x_j}(.) = k(x_j, .)$ are members of $\mathcal{H}$ then according to the second property of the RKHS $\mathcal{H}$,

$$\langle k(x_i, .), k(x_j, .) \rangle_{\mathcal{H}} = k(x_i, x_j) \ .$$

Therefore, the solution of the integral in eq. (2.11) relies on it being a particular case of an inner product between two reproducing members of $\mathcal{H}$ and is as simple as a function evaluation of k.

SE hyperparameters The SE is a stationary kernel, i.e. it is a function of distance $d = x_i - x_j$ which makes it *translation invariant* (in time). It is governed by the characteristic lengthscale ℓ^2 which, roughly speaking, specifies the distance at which the outputs for any two inputs (x_i, x_j) become uncorrelated. In other words, the lengthscale ℓ^2 controls the amount by which f varies along the input domain (time): A small lengthscale makes f vary rapidly along time, and a very large lengthscale makes f behaves almost like a constant function, see Figure 2.2. This parameterisation of the SE kernel is very powerful when combined with hyperparameter *adaptation*, as described in section 2.2.4. Other adaptable hyperparameters include the signal variance σ_f^2 which is the vertical scale of function variation and the noise variance σ^2, see eq. (2.2). The noise variance is not a hyperparameter of the SE itself, but of its noisy variant. Unless we set it as a constant, its adaptation can give *different explanations* about the latent function that generates the data.

Kernel composites One can also combine covariance functions, as long as they are positive-definite. In fact, eq. (2.12) is a sum of the SE kernel and the covariance function of isotropic Gaussian noise. Examples of valid combinations of covariance functions include *linear combinations* and *products* of covariance functions. *Direct sums* and *tensor products* of covariance functions defined over different spaces are also valid covariance functions.

2.2.3 Gaussian process regression

To reconstruct the true trajectory of gene expression at the sampled inputs (timepoints) as well as predict the trajectory at unsampled inputs we must infer the true function values $f(\mathbf{x})$, as well as $f(x_*)$ for all new inputs x_*, given some observed outputs $\mathbf{y}$ at sampled inputs $\mathbf{x}$[1].

Under the GP model in eq. (2.7), we know that the joint distribution over any (latent) function values $\mathbf{f} = f(\mathbf{x})$ is the *GP prior*. Without loss of generality, $\mathbf{f}$ can be concatenated with a single new function value f_* for some unsampled input x_*,

$$\begin{bmatrix} \mathbf{f} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_f & \mathbf{k}_{f_*} \\ \mathbf{k}_{f_*}^\top & k_* \end{bmatrix}\right) ,$$

where $\mathbf{K}_f$ is the RBF covariance matrix across the outputs of sampled timepoints $\mathbf{x}$, $(\mathbf{k}_{f_*})_i = k(x_i, x_*)$ is the covariance between the new f_* and old function values $\mathbf{f}$ and $k_* = k(x_*, x_*)$ is the variance of f_*.

Noisy outputs As with any practical application, in this section we consider predictions using noisy observations $\mathbf{y}$, whereas the true function values $\mathbf{f}$ are unknown. Since the noise is spherical Gaussian by assumption, then

$$\operatorname{cov}\begin{bmatrix} \mathbf{y} \\ f_* \end{bmatrix} = \operatorname{cov}\begin{bmatrix} \mathbf{f} & + \boldsymbol{\epsilon} \\ f_* & + 0 \end{bmatrix} = \operatorname{cov}\begin{bmatrix} \mathbf{f} \\ f_* \end{bmatrix} + \operatorname{cov}\begin{bmatrix} \boldsymbol{\epsilon} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{K}_f + \sigma^2\mathbf{I} & \mathbf{k}_{f_*} \\ \mathbf{k}_{f_*}^\top & k_* \end{bmatrix} , \qquad (2.13)$$

[1] Where possible, we do not denote the conditioning on inputs to avoid cluttering.

so

$$\begin{bmatrix} \mathbf{f} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_f & \mathbf{k}_{f_*} \\ \mathbf{k}_{f_*}^\top & k_* \end{bmatrix}\right) , \tag{2.14}$$

$(\mathbf{y}^\top \ f_*)$ is Gaussian-distributed with zero mean and the covariance in eq. (2.13). Now, the closed form of the predictive distribution of $f_* \,|\, \mathbf{y}$ relies on standard formulas for the conditional mean and covariance of a subset of Gaussian random variables conditioned on the rest, see appendix A.1.

Predictive equations

The mean and covariance of the predictive distribution of $f_* \,|\, \mathbf{y}$ define the mean function and covariance function of the *posterior* GP, which can be seen intuitively as a distribution over functions that agree with our observations $(\mathbf{x}, \mathbf{y})$, see Figure 2.2(a). For a single new timepoint x_* we have:

$$f_* \,|\, \mathbf{y} \sim \mathcal{N}\left(m_*, \mathrm{var}\left[f_*\right]\right) \ , \quad \text{where}$$

$$m_* = \mathbf{k}_{f_*}^\top (K_f + \sigma^2 \mathbf{I})^{-1} \mathbf{y} \ , \tag{2.15}$$

$$\mathrm{var}\left[f_*\right] = k(x_*, x_*) - \mathbf{k}_*{}^\top (K_f + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_* \ . \tag{2.16}$$

These equations can be easily generalised for the prediction of function values for a set of new timepoints $\mathbf{x}_*$, by augmenting $\mathbf{k}_{f_*}$ with more columns (one for each new timepoint $\mathbf{x}_*$) and turning $k(x_*, x_*)$ into the matrix defined by $(\mathbf{K}*)_{ij} = k(x_{*i}, x_{*j})$.

With respect to the joint covariance in eq. (2.13), for every new timepoint x_*, a new vector $\mathbf{k}_{f_*}$ is concatenated as an additional column and row to give

$$\mathbf{K}_{C+1} = \begin{bmatrix} \mathbf{K}_C & \mathbf{k}_{f_*} \\ \mathbf{k}_{f_*}^\top & k_* \end{bmatrix} ,$$

where C increments with every new timepoint.

2.2.4 Hyperparameter learning

Given the data, one can learn the hyperparameters of the kernel by maximising the marginal likelihood of the GP $p(\mathbf{y} \,|\, \mathbf{x}, \boldsymbol{\theta})$, that is, the marginal distribution over outputs $\mathbf{y}$, governed by the hyperparameters $\boldsymbol{\theta}$ of $\mathbf{K}_y$, see eq. (2.12). In general, a kernel-based model such as the GP can employ a variety of kernel families whose hyperparameters can be adapted with respect to the underlying intensity and frequency of the local signal structure. The GP can then predict the true signal while quantifying the uncertainty of the prediction, that is, the signal reconstruction happens in a probabilistic fashion. The SE kernel allows us to interpret the adapted hyperparameters intuitively, especially for one-dimensional inputs such as time-series, see Figure 2.2 for an example of interpreting various local optima.

Maximising the GP marginal likelihood

We get a closed form of the marginal likelihood of the GP model by marginalising over the latent function values $\mathbf{f}$:

$$p(\mathbf{y} \,|\, \mathbf{x}) = \int \mathrm{d}\mathbf{f} p(\mathbf{y} \,|\, \mathbf{f}, \mathbf{x}) p(\mathbf{f} \,|\, \mathbf{x}) = \mathcal{N}\left(\mathbf{y} \,|\, \mathbf{0}, \mathbf{K}_f + \sigma^2 \mathbf{I}\right) \ , \tag{2.17}$$

where $p(\mathbf{f} \,|\, \mathbf{x})$ is the GP prior from eq. (2.7) and $p(\mathbf{y} \,|\, \mathbf{f}, \mathbf{x})$ is the Gaussian likelihood $\mathcal{N}(\mathbf{y} \,|\, \mathbf{f}, \sigma^2 \mathbf{I})$ factorised across the outputs $\mathbf{y}$. It is common to compute the *log* of the marginal likelihood (LML), as it is more stable numerically and prevents arithmetic underflows:

$$\ln p(\mathbf{y} \,|\, \mathbf{x}, \boldsymbol{\theta}) = -\tfrac{1}{2} \mathbf{y}^\top \mathbf{K}_y^{-1} \mathbf{y} - \tfrac{1}{2} \ln|\mathbf{K}_y| - \tfrac{n}{2} \ln(2\pi) \ , \tag{2.18}$$

where $\mathbf{K}_y = \mathbf{K}_f + \sigma^2 \mathbf{I}$. Note that the marginal here is explicitly conditioned on the hyperparameters $\boldsymbol{\theta}$ to denote it as a function of the hyperparameters of $\mathbf{K}_f$.

To maximise the marginal likelihood, we use the matrix derivative identities in appendix A.2 to compute partial derivatives of the LML with respect to each hyperparameter:

$$\begin{aligned}\tfrac{\partial}{\partial\theta}\ln p(\mathbf{y}\,|\,\mathbf{x},\,\boldsymbol{\theta}) &= \tfrac{1}{2}\boldsymbol{\alpha}^{\top}\tfrac{\partial}{\partial\theta}\mathbf{K}_y\,\boldsymbol{\alpha} - \tfrac{1}{2}\mathrm{tr}\left(\mathbf{K}_y^{-1}\tfrac{\partial}{\partial\theta}\mathbf{K}_y\right)\\ &= \tfrac{1}{2}\mathrm{tr}\left(\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^{\top} - \mathbf{K}_y^{-1}\right)\tfrac{\partial}{\partial\theta}\mathbf{K}_y\right)\ ,\end{aligned}$$

for $\boldsymbol{\alpha} = \mathbf{K}_y^{-1}\,\mathbf{y}$ and

$$\begin{aligned}\tfrac{\partial}{\partial\ell^2}k_y(x_i,x_j) &= k_f(x_i,x_j)\frac{(x_i-x_j)^2}{2\ell^4}\ , & \tfrac{\partial}{\partial\ell^2}\mathbf{K}_y &= \mathbf{K}_f\circ\tfrac{1}{2\ell^4}\mathbf{D}\ ,\\ \tfrac{\partial}{\partial\sigma_f^2}k_y(x_i,x_j) &= k_f(x_i,x_j)\ \sigma_f^{-2}\ , & \tfrac{\partial}{\partial\sigma_f^2}\mathbf{K}_y(x_i,x_j) &= \tfrac{1}{\sigma_f^2}\mathbf{K}_f\ ,\\ \tfrac{\partial}{\partial\sigma^2}k_y(x_i,x_j) &= \delta_{ij}\ , & \tfrac{\partial}{\partial\sigma^2}\mathbf{K}_y &= \mathbf{I}\ ,\end{aligned}$$

where $\circ$ denotes the Hadamard product and $(\mathbf{D})_{ij} = (x_i - x_j)^2$ is the matrix of squared differences. The LML can be optimised through the *scaled conjugate gradients* algorithm [Möller, 1993], to which we feed the partial derivatives listed above.

2.2.5 Model comparison and ranking with likelihood ratios

Maximising the LML is fundamentally a maximum-likelihood approach, known as type II maximum-likelihood[1]. Alternatively, we could opt for a fully Bayesian approach, by assuming a *hyper-prior* distribution $p(\boldsymbol{\theta}\,|\,\mathcal{M})$ over the hyperparameters, where $\mathcal{M}$ represents a particular class of models. The posterior over the hyperparameters,

$$p(\boldsymbol{\theta}\,|\,\mathbf{y},\mathbf{x},\mathcal{M}) = \frac{p(\mathbf{y}\,|\,\mathbf{x},\boldsymbol{\theta},\mathcal{M})\ p(\boldsymbol{\theta}\,|\,\mathcal{M})}{\int \mathrm{d}\boldsymbol{\theta}\ p(\mathbf{y}\,|\,\mathbf{x},\boldsymbol{\theta},\mathcal{M})\ p(\boldsymbol{\theta}\,|\,\mathcal{M})}\ , \tag{2.19}$$

would be based on some initial beliefs encoded in $p(\boldsymbol{\theta}\,|\,\mathcal{M})$, such as the functions having large lengthscales. A maximum a posteriori (MAP) approach would amount to promoting large lengthscale values (via the prior term), unless there

[1] As opposed to type I maximum-likelihood on the data likelihood $p(\mathbf{y}\,|\,\mathbf{x},\mathbf{f})$. Type II optimises the parameters of a marginal model.

is evidence to the contrary (via the likelihood term). The normalising constant in the denominator is known as the *model evidence*.

In the presence of different classes of models $(\mathcal{M}_1, \mathcal{M}_2)$, that is, with a different set of hyperparameters $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$, a Bayesian-standard way of comparing them is through:

$$R(\mathcal{M}_1, \mathcal{M}_2) = \frac{p(\mathbf{y} \,|\, \mathbf{x}, \mathcal{M}_1)}{p(\mathbf{y} \,|\, \mathbf{x}, \mathcal{M}_2)} \times \frac{p(\mathcal{M}_1)}{p(\mathcal{M}_2)} = \frac{p(\mathcal{M}_1 \,|\, \mathbf{y}, \mathbf{x})}{p(\mathcal{M}_2 \,|\, \mathbf{y}, \mathbf{x})} \,. \tag{2.20}$$

The first ratio in the RHS is the *Bayes* factor — a ratio of model evidence terms (recall the one that appears in the denominator of eq. (2.19)), where the models $(\mathcal{M}_1, \mathcal{M}_2)$ usually represent two complementary hypotheses. Namely,

- $\mathcal{M}_1$ - *the profile has a significant underlying signal and thus it is truly differentially expressed.*
- $\mathcal{M}_2$ - *there is no underlying signal in the profile and the observed gene expression is simply the effect of random noise.*

In other words, if we can compute this ratio then we can rank the profiles based on how likely it is that model $\mathcal{M}_1$ generated the data relative to $\mathcal{M}_2$. The second factor is a ratio of model priors which weighs the Bayes ratio based on our initial beliefs about the models. Again, this turns out to be a trade-off between initial belief (expert, domain or simply gut knowledge) and empirical evidence — a recurring theme of Bayesian reasoning. Usually, as is the case in here, if there is no good reason to believe that any one model is more probable, then a uniform $p(\mathcal{M})$ is used. See also [Angelini *et al.*, 2007; Stegle *et al.*, 2010; Yuan, 2006] for other examples of hypotheses comparisons within a Bayesian framework.

In practice, the model class $\mathcal{M}$ is such that the integral (model evidence) in eq. (2.19) is analytically intractable. Standard approaches to approximating the posterior distribution include the Laplace approximation, or sampling from the posterior with Markov chain Monte Carlo (MCMC) methods to discover its — potentially multiple — modes [MacKay, 1999; Neal, 1997].

Approximating ratio In this case study we present a simpler but effective approach to ranking the differential expression of a profile: Instead of integrating out the hyperparameters, we approximate the Bayes factor with a log-ratio of GP marginal likelihoods (introduced in eq. 2.18):

$$R(\mathcal{M}_1, \mathcal{M}_2) \approx \ln\{p(\mathbf{y} \,|\, \mathbf{x}, \boldsymbol{\theta}_1)\} - \ln\{p(\mathbf{y} \,|\, \mathbf{x}, \boldsymbol{\theta}_2)\}, \tag{2.21}$$

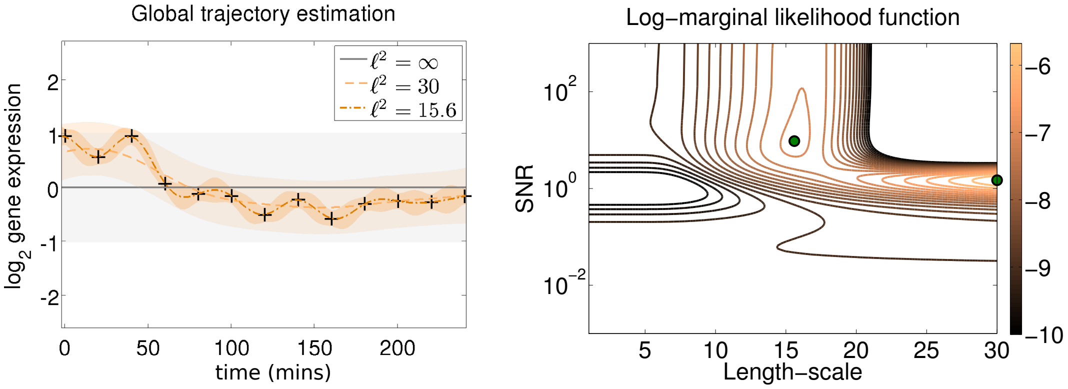
with each LML being a function of different instantiations of $\boldsymbol{\theta}$. That is, we still maintain hypotheses $\mathcal{M}_1$ and $\mathcal{M}_2$, representing the same notions as described above, but in our case they differ simply by configurations of $\boldsymbol{\theta}$.

Hyperparameter configuration Specifically, with $\mathcal{M}_2$ the hyperparameters are fixed to $\boldsymbol{\theta}_2 = (\infty, 0, \hat{\text{var}}[y])^\top$ to encode a function constant in time $[\ell^{\,2} \to \infty]$, with no underlying signal $[\sigma_f^2 = 0]$, which generates a time-series with an empirical variance that is explained exclusively by noise $[\sigma^2 = \hat{\text{var}}[y]]$. Analogously, with $\mathcal{M}_1$ the hyperparameters $\boldsymbol{\theta}_1 = (20, \hat{\text{var}}[y], 0)^\top$ are initialised in a way that encodes a function fluctuating in accordance to a typical significant profile — for instance $\ell^2 = 20$ — with a signal variance that exclusively explains the empirical time-series variance $[\sigma_f^2 = \hat{\text{var}}[y]]$ and no noise $[\sigma^2 = 0]$.

Local optima of the LML function

The two configurations $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ correspond to two points in the three-dimensional input domain of the LML function, both of which usually lie close to local-optimum solutions. This assumption can be empirically verified by exhaustively plotting the LML function for many instantiations of $\boldsymbol{\theta}$, see Figure 2.2(b). For the less frequent case of profiles whose LML contour varies radically, a number of initialisation points can be used to ensure convergence to the global-maximum solution. Because the configuration of the first hypothesis (no noise, $\sigma^2 = 0$) is an unrealistic scenario, we let $\boldsymbol{\theta}_1$ adapt with respect to a given profile by numerically optimising its LML function, as opposed to keeping it fixed like $\boldsymbol{\theta}_2$.

In general, the LML as a function of $\boldsymbol{\theta}$, eq. (2.18), is not convex. Nonetheless,



(a)

(b)

Figure 2.2: (a) GP fit on the centred profile of gene `Cyp1b1` (probeID `1416612_at` in the `GSE10562` dataset) with different settings of the lengthscale hyperparameter ℓ^2. The crosses are zero-mean hybridised gene expression (log2 ratios between treatment and control). The solid and dotted lines are mean predictions of the GP and the shaded areas visualise the point-wise mean -/+ two standard deviations (95% confidence region). As $\ell^2 \to \infty$ (0 inverse-lengthscale), the mean function becomes virtually constant and the empirical output variance is attributed to noise, $\hat{\text{var}}[y] = \sigma^2$. When the lengthscale ℓ^2 is set to a large enough value (local-optimum $\ell^2 = 30$), the mean function roughly fits the data-points and the observed variance is explained equally by signal and noise, ($\sigma_f^2 = \sigma^2 = 2\ \hat{\text{var}}[y]$). Additionally, the GP has higher uncertainty in its predictive curve. When the lengthscale is set to a local-optimum of a small lengthscale ($\ell^2 = 15.6$) then the mean function tightly fits the data-points with high certainty. The interpretation from the covariance function in this case is that the profile contains a minimal amount of noise and that most of the empirical output variance is explained by the underlying signal, $\hat{\text{var}}[y] = \sigma_f^2$. (b) The contour of the corresponding LML function plotted through an exhaustive search of ℓ^2 and SNR (signal-to-noise ratio) values. The two main local-optima are indicated with green dots and a third local optimum, that corresponds to the constant zero function, has a virtually flat vicinity in the contour, which encompasses the whole lengthscale axis for very small values of SNR (that is, the lengthscale is not important when SNR≈ 0).

local optima do not necessarily pose an obstacle to learning, but provide alternative interpretations to the observations. However, to help alleviate the problem of spurious local optimum solutions, we make the following observation: by explicitly restricting the signal variance hyperparameter σ_f^2 to small values during optimisation, we implicitly restrict the noise variance hyperparameter σ^2 to large values. This occurs as the explanation of the empirical output variance $\hat{\text{var}}[y]$ is split between the underlying signal and noise variance, that is, $\hat{\text{var}}[y] = \sigma_f^2 + \sigma^2$. This dependency allows us to treat this three-dimensional optimisation problem as an intrinsically two-dimensional problem — of a lengthscale ℓ^2 dimension and of a SNR (signal-to-noise ratio) $= \sigma_f^2/\sigma^2$ dimension — without danger of missing any optima.

Figure 2.2(b) illustrates the log-marginal likelihood as a function of the characteristic lengthscale ℓ^2 and the SNR. It features two local optima, one with a small lengthscale and high SNR, where the observed data are explained with a complex function and small noise variance, and one optimum for a large lengthscale and a low SNR, where the data are explained by a simpler function with high noise variance. Note that the first optimum has a smaller LML. This relates to the algebraic structure of the LML, eq. (2.18): the first term (dot product) promotes data fitness and the second term (determinant) penalises the complexity of the model [Rasmussen & Williams, 2006, sec.5.4].

Overall, the LML function of the Gaussian process offers a good trade-off between fitness and complexity without the need for additional regularisation. Optionally, we can use multiple initialisation points that focus on different finite lengthscales, to deal with the local optima along the lengthscale axis. Finally, we pick the best solution (max LML) to represent hypothesis $\mathcal{M}_1$ in the likelihood-ratio during the ranking stage.

2.3 Results and discussion

We apply standard GP regression and the Bayesian hierarchical model for the analysis of time-series (BATS) on two in-silico datasets simulated by BATS and GPs, and on one experimental dataset coming from a study on primary mouse keratinocytes with an induced activation of the TRP63 transcription factor [Della Gatta *et al.*, 2008]. In that study, a reverse-engineering algorithm was developed (TSNI: time-series network identification) for inferring the direct targets of TRP63.

2.3.1 Evaluation setup

ROC curves We assume that each gene expression profile can be categorized as either quiet or differentially expressed. We consider algorithms that provide a ranking of the profiles, on the basis of which is most likely to be non-quiet (or differentially expressed). Given a ground truth, we can then evaluate the quality of such a ranking and compare different algorithms. We use *receiver operating characteristic* (ROC) curves to evaluate the algorithms. These curves plot the *false positive rate* on the horizontal axis, versus the *true positive rate* on the vertical axis; that is, the percentage of negatives (non-differentially expressed profiles in the ground truth) that were erroneously declared positive (declared differentially expressed), versus the percentage of positives (in the ground truth) that were correctly declared as positives.

Ground truths In this case study, we consider a ground truth to consist of a binary vector, of equal length to the number of profiles in the dataset, where the label "1" flags the corresponding profile as differentially expressed and the label "0" as non-differentially expressed. This can be viewed as a binary classification problem, with a threshold on the ranking-metric playing the role of the decision boundary. By varying that threshold, the corresponding points (FPR, TPR) form an ROC curve. The quality of a ranking can then be summarised by the AUC of the corresponding curve. So a good ranking exhibits a rapidly rising percentage of its first i positions having matching labels [1/0] to the ground truth, as i increases from 1 to N (all profiles). The following subsections (2.3.2, 2.3.3) describe three such ground truths in detail.

2.3.2 Simulated data

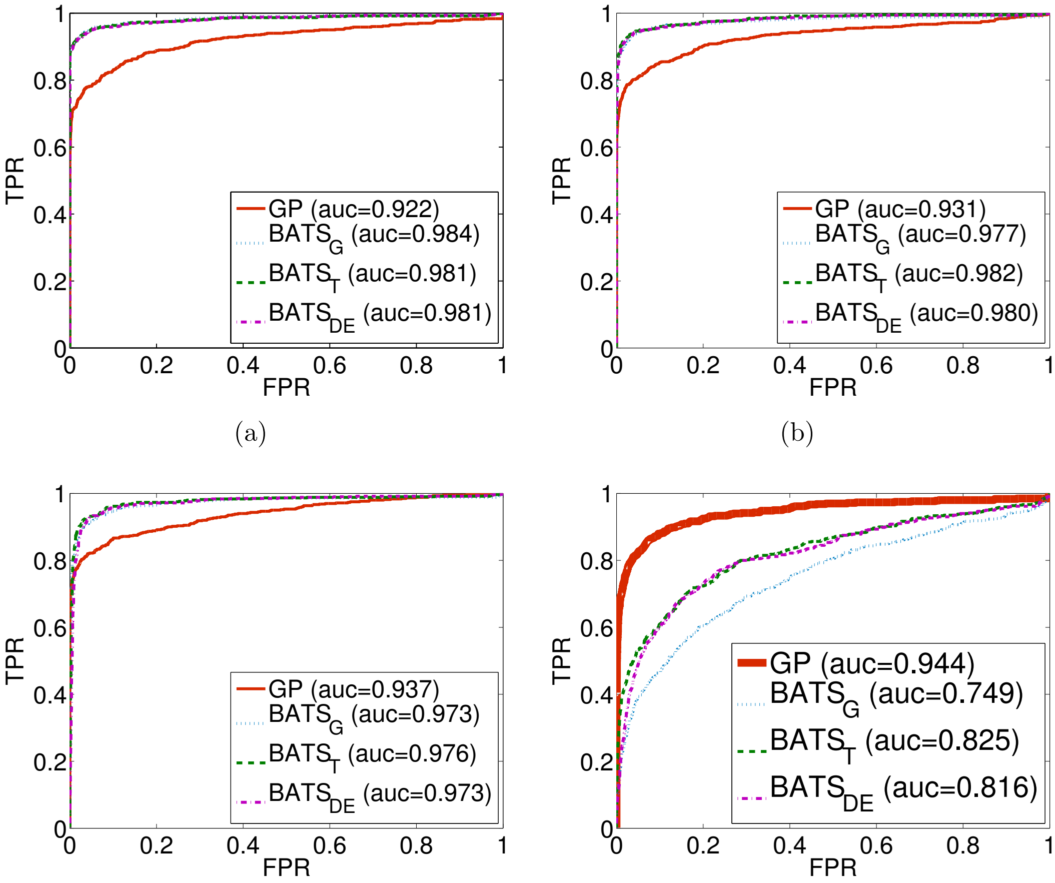
Bayesian Analysis of Time Series In BATS [Angelini *et al.*, 2007], the assumption is that each time-course profile is generated by a function projected on an orthonormal basis (Legendre or Fourier), plus some noise. Thus the global estimand for every gene expression trajectory is the linear combination of a number of basis functions, whose coefficients are modeled by a posterior distribution. The number of basis functions and their coefficients, are estimated with closed form computations in a fully Bayesian manner. The BATS framework also allows for various types of non-Gaussian noise models.

BATS simulation

The first set of in-silico profiles is simulated by the BATS software[1] in accordance to the guidelines given by Angelini *et al.* [2008]. We reproduce the simulations performed by Angelini *et al.* [2007]. Specifically, we sample three sets of $N = 8000$ profiles, with $n = 11$ timepoints and $k_i^j = 2$ replicates, for $i = 1, \ldots, N$, $j = 1, \ldots, n$ except $k_i^{2,5,7} = 3$, according to the model defined in [Angelini *et al.*, 2007, sec. 2.2]. In each of the three sets of profiles, 600 out of 8000 are generated as differentially expressed (labeled “1” in the ground truth), that is, they are simulated as a linear combination of orthonormal basis function (Legendre polynomials) with additive iid noise.

The other 7400 non-differentially expressed profiles (labeled as “0” in the ground truth) are zero functions with additive iid noise. Each BATS-sampled dataset is induced with a different kind of iid noise — Gaussian $N(0, \sigma^2)$, Student-t distributed with 5 and 3 degrees of freedom (T(3), T(5)). Figure 2.3(a,b,c) illustrates the comparison on BATS-sampled data with various kinds of noise.

[1] http://www.na.iac.cnr.it/bats/



(a)

(b)

(c)

(d)

Figure 2.3: ROC curves for the GP and BATS methods on data simulated by BATS induced with (a) Gaussian noise, (b) Student's-t with 5 degrees of freedom, (c) Student's-t with 3 degrees of freedom, (d) data simulated by GPs. Each panel depicts one ROC curve for the GP method and three for BATS, each using a different noise model indicated by the subscript legend (**G**aussian, Student's-**T** and **D**ouble-**E**xponential), followed by the AUC.

Gaussian process simulation

In a similar setup, the second in-silico dataset consists of $N = 8000$ profiles sampled from GPs, with the same number of replicates and timepoints, among which 600 are differentially expressed. To generate a differentially expressed profile, we first sample the *hyperparameters* of the RBF kernel from separate Gamma densities, one for each each hyperparameter. To resemble the behaviour of BATS-sampled profiles, each Gamma density is fitted to a set of estimates of the corresponding hyperparameter. This is a set of estimates taken from GP training on BATS-sampled profiles. The set only includes estimates from TP-classified profiles at FPR ≈ 0. Table 2.1 lists the fitted parameters of the Gamma densities.

		Sampling Gamma density $\Gamma(a, b)$	
		a (scale)	b (shape)
Sampled RBF-hyperparameter	ℓ^2 (lengthscale)	1.4	5.7
	σ_f^2 (signal variance)	2.76	0.2
	σ^2 (noise variance)	23	0.008

Table 2.1: Gamma distributions from which we sample the RBF hyperparameters. For instance, σ_f^2 is sampled from a Gamma with scale 1.4 and shape 5.7. The hyperparameters are then used in the RBF covariance function to simulate a profile from the GP.

The other 7400 non-differentially expressed profiles are simply zero functions plus iid spherical Gaussian noise, with variance equal to the sum of a sampled σ_f^2 and σ^2, in accordance to Table 2.1. This generates a noise-only profile of comparative amplitude to the differentially expressed ones. Figure 2.3(d) illustrates the comparison on the GP-sampled data.

2.3.3 Experimental data

We apply the standard GP regression framework and BATS on an experimental dataset[1] from a study on primary mouse keratinocytes with an induced activation

[1] GEO database accession number GSE10562.

of the TRP63 transcription factor [Della Gatta *et al.*, 2008]. In this study the authors developed TSNI (Time-Series Network Identification), a reverse-engineering algorithm for inferring the direct targets of TRP63. Based on the AUC of the gene expression trajectories, 786 out of 22690 gene reporters were chosen and ranked by TSNI according to the probability of being direct targets of TRP63. This ranking list[1] serves here as a *noisy ground truth*.

Preprocessing Prior to any analysis, we process the data with the RMA (robust multi-array average) expression measure, built in the `affy` R-package [Irizarry *et al.*, 2003]. We label as "1" the top 100 position of the TSNI ground truth ranking, as they are the best candidate direct targets of the TRP63 transcription factor. This is justified in Figure 2.4, where the distribution of the *binding scores*[2] is denser within the first 100 ranks. Furthermore, Della Gatta *et al.* [2008] validated these 100 positions via GSEA (gene set enrichment analysis) [Subramanian *et al.*, 2005] to correlate their up/down regulation patterns to genes that respond to TRP63 knock-downs in general. In summary, *"the top 100 TSNI ranked transcripts are significantly enriched for the strongest binding sites"* [Della Gatta *et al.*, 2008]. Figure 2.5 illustrates the comparison on the experimental data.

2.3.4 Comparison

Ultimately, each model outputs a ranking of differential expression which is assessed by an ROC curve to quantify how well in accordance to each of three ground truths [BATS-sampled, GP-sampled, TSNI-experimental] the method performs. The BATS model uses three different noise models, that is, the marginal distribution of the error is assumed to be either Gaussian, Student-t or double-exponential. For the following comparisons we plot four ROC curves, one for each noise model of BATS and one for the GP. We demonstrate that the GP ranking

[1] Published as a supplementary file: http://genome.cshlp.org/content/suppl/2008/05/05/gr.073601.107.DC1/DellaGatta_SupTable1.xls

[2] Computed as the sum of -log2 of *p-values* of all TRP63-binding regions identified by ChIP-chip experiments.

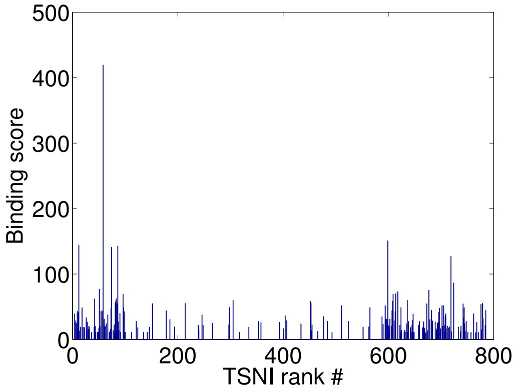


Figure 2.4: The distribution of the binding scores is mostly dense along the first 100 positions of the TSNI ranking. Della Gatta *et al.* [2008] selected only the top 100 and bottom 200 genes to look for binding sites and thus showed that the top 100 genes have more binding sites than the bottom 200 genes. The limited concentration in the between-ranks is due to some regions along the genome being occupied by the same reporter in the microarray.

outperforms that of BATS with respect to the TSNI ground truth ranking on the experimental data (Figure 2.5) and, as expected, on GP-sampled profiles (Figure 2.3(d)).

2.3.5 Discussion

On BATS-sampled data, Figure 2.3(a,b,c), we observe that the change in the induced noise is barely noticeable in regards to the performances of both methods and that BATS maintains its stable supremacy over the GP framework. This performance gap is partially due to the lack of a robust noise model for the GP (see section 2.4.1, **Related work**). Furthermore, there is a modeling bias in the underlying functions of the simulated profiles, which contain a finite small degree of differentiability[1]. This puts the GP in a disadvantaged position as it models for smooth (infinitely differentiable) functions due to its *squared exponential* covariance function. Consequently, for this simulated dataset the GP is more susceptible to capturing spurious patterns as they are more likely to lie within

[1] The maximum degree of Legendre polynomials is 6.

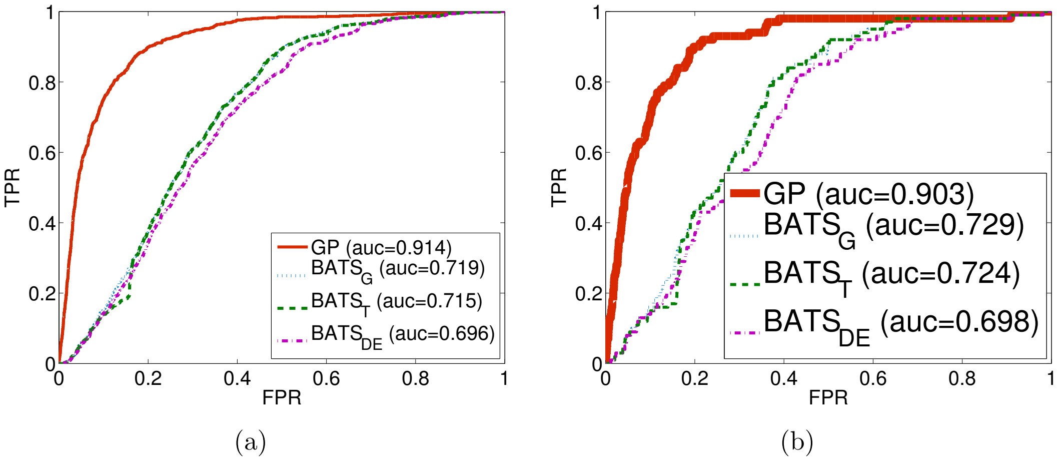


Figure 2.5: ROC curves for the GP and BATS methods on the experimental data. As with the simulated data, one ROC curve and its AUC are depicted for the GP method and three for BATS, each using a different noise model indicated by the subscript in the legend. **(a)** Ground truth consists of 22690 labels among which only 786 profiles, top-ranked by TSNI, are labeled “1”. **(b)** Similarly, here only 100 profiles, top-ranked by TSNI, are labeled “1”.

its modeling range, whereas for BATS modeling the polynomials with a limited degree acts as a safeguard against spurious patterns, most of which vary rapidly in time.

On GP-sampled data, Figure 2.3(d), we observe a switch in terms of superiority in favor of the GP framework, while its performance is virtually unaffected. The GP still seems susceptible to non-differentially expressed profiles with spurious patterns as well as differentially expressed profiles with excessive noise. However, the polynomials of limited degree of BATS show to be inadequate for many of the GP-sampled functions and the two BATS variants with robust noise models (BATS$_T$, BATS$_{DE}$) only alleviate the problem slightly.

Similarly, Figure 2.5 shows the GP maintaining superiority over the Gaussian noise variant of BATS by a similar degree. The experimental data are more complex and the robust BATS variants seem to offer no performance boost. Since the ground truth focuses on the 100 most differentially expressed genes, with respect to the induction of the TRP63 transcription factor, then these results indicate that the proposed GP ranking method indeed highlights differentially expressed genes, with an attractive robustness against various kinds of noise.

2.4 Conclusions

We presented an approach to estimating the continuous trajectory of gene expression time-series from microarray data through Gaussian process regression and ranking the differential expression of each profile through a log-ratio of two GP marginal likelihoods, each one representing the hypothesis of differential and non-differential expression respectively.

We compared our method to a recent Bayesian hierarchical model (BATS) via ROC curves, on data simulated by BATS and GPs and experimental data. The experimental data were taken from a previous study on primary mouse keratinocytes and the top 100 genes of its ranking were used here as the noisy ground truth for the purposes of assessment.

The GP framework significantly outperforms BATS on experimental and GP-sampled data and the results show that standard GP regression can be regarded as a standard tool in evaluating the continuous trajectories of gene expression and ranking its differential expression.

One of our primary assumptions in this chapter was that of an unstructured noise process. Once we explained any structure with the RBF, all that was left was iid Gaussian spherical noise. In chapter 3 we present a framework for uncovering any structured noise, given a partial explanation of the joint covariance. Later in chapter 5 we will demonstrate this idea of residual analysis as a sequel to the analysis in this chapter.

2.4.1 Related work

The proposed ranking scheme relates to the work of Stegle *et al.* [2010] on *two-sample* data (separate time-course profiles for each treatment), where the two competing hypotheses are represented by two different generative models connected by a *gating* scheme: one hypothesis assumes that the two profiles of a gene reporter are generated by two different GPs, explaining the gene as *differentially expressed* across the two treatments. The other hypothesis assumes that the two profiles are generated by the same GP, thus the gene is *non-differentially expressed*. The *gate* serves as a switch between the two generative models, in time, to detect *intervals* of differential expression. This gives biologists a means for investigating the propagation of perturbations in a gene regulatory network.

Practicalities aside, this case study demonstrates that Gaussian process regression is a natural fit to the analysis of gene expression time-series and its simplicity can still outweigh the ever-increasing, but necessary, complexity of hierarchical Bayesian models.

2.4.2 Future work

While this case study and the proposed methodology follow a more basic approach, we note that robust mechanisms against outliers, such as the ones used by Stegle *et al.* [2010] (see also Tipping & Lawrence, 2005; Vanhatalo *et al.*, 2009), are complementary to this work and including one would be a sensible extension of our framework. Finally, the potential periodicity of the underlying signal sets another interesting biological question about the behaviour of gene expression. For this purpose, a different kind of stationary covariance function, the *periodic* covariance function [MacKay, 2003, section 45.4], can be used to fit a time-series generated by a periodic process, with the lengthscale hyperparameter interpreted as its cycle.

2.4.3 Source code

The source code for the GP regression framework is available in Matlab code[1] and as a package for the R statistical language[2]. The routines for the estimation

[1] http://staffwww.dcs.shef.ac.uk/people/N.Lawrence/gp/

[2] http://cran.r-project.org/web/packages/gptk/

and ranking of the gene expression time-series are available in Matlab[1] and as an R Bioconductor package[2]. The time needed to serially analyse the 22690 profiles in the experimental dataset, with just the two basic initialisation points of hyperparameters, is roughly 30 minutes on a desktop running Ubuntu 10.04 with a 2.8GHz CPU and 3.2 GiB of memory. Since the gene expression profiles are independently fitted, the procedure can be parallelised for N cores, cutting the computation time down to 30/N minutes.

2.4.4 Authors contributions

Alfredo Kalaitzis (AK) designed and implemented the computational analysis and ranking scheme presented here, assessed the various methods and drafted the related manuscript. Neil Lawrence (NL) pre-processed the experimental data and wrote the original Gaussian process toolkit for MATLAB and AK rewrote it for the R statistical language. Both AK and NL participated in interpreting the results and revising the manuscript.

[1] http://staffwww.dcs.shef.ac.uk/people/N.Lawrence/gprege/

[2] http://www.bioconductor.org/packages/2.10/bioc/html/gprege.html

Chapter 3

Residual component analysis

One of our primary assumptions in chapter 2 was that of an unstructured noise process. Once we explained any structure with the RBF, the residuals were iid Gaussian spherical noise. In this chapter we study scenarios where the noise has structure and present the *residual component analysis* (RCA) framework that generalises (probabilistic-)PCA for recovering such complex residuals when a partial explanation of the joint covariance is given. Later in chapter 5 we will demonstrate this idea of RCA in the context of the regression problem from chapter 2.

Probabilistic principal component analysis (PPCA) seeks a low dimensional representation of a data set in the presence of independent spherical Gaussian noise. The maximum-likelihood solution for the model is based on an eigenvalue problem on the sample-covariance matrix. In this chapter we consider the situation where the data variance is already *partially* explained by other factors. For instance, these can be *sparse conditional dependencies* between the covariates, or *temporal correlations* between datapoints in a time-series; ultimately, these factors leave some *residual variance* unexplained.

We address the problem of decomposing *only* the residual variance into its eigenvector components through a *generalised eigenvalue problem* (GEP), which we call *residual component analysis* (RCA) [Kalaitzis & Lawrence, 2011a, 2012]. We explore a range of new algorithms that arise from the framework, including one that decomposes the covariance of a Gaussian distribution into a low-rank and a sparse-inverse component. We show that *principal component analysis* (PCA), *canonical correlation analysis* (CCA) and *linear discriminant analysis* (LDA) can be derived as special cases of our algorithm. Furthermore, we discuss a deeper connection of these methods on the basis of oblique (non-orthogonal) projections steered by the structure of the explained covariance term.

Roadmap

We start by giving some background on *probabilistic principal component analysis* (PPCA) in section 3.1, a Gaussian model that provides a probabilistic interpretation of a classical linear approach on dimensionality reduction. In the same section we also discuss *dual*-PPCA, a basic linear model that is the dual counterpart of PPCA (that is, it describes the relationships between *datapoints* as opposed to *features*). This will lay the basis for any *kernel*-based approach that we might attempt to devise in the future.

The shortcomings of the simplistic low-rank plus diagonal covariance will become clear while describing some useful manifestations of linear mixed-effects models, in section 3.2. This will motivate the use of a more general, *low-rank plus positive definite*, covariance structure along with an algorithm for learning the low-rank component when the positive definite term is known or estimated. In section 3.3 this idea will crystallise in the form of a formal proof on maximising the likelihood of this covariance structure with respect to the low-rank part.

Aside from our contribution in endowing this structure with a probabilistic interpretation, termed *residual component analysis* (analogous to the PPCA-PCA relationship), the task of explaining away structure with some fixed covariates in linear models has long been explored in the statistics and signal processing literature. Therefore, we also aim to uncover a deep connection between any problem that can be cast as a PCA problem (sec. 3.3.3), their probabilistic counterparts and oblique projections (sec. 3.3.2).

At that point, we will have justified RCA as a probabilistic *model* that unifies many different algorithms (chapter 4), as opposed to it being a mere algorithmic trick on an eigenvalue problem. This will lay the basis for using RCA in larger graphical models and potentially formulate Bayesian extensions (for example, with sparsity priors on the loadings) or kernelised generalisations through the dual version of the RCA theorem. While this chapter focuses on the theory of RCA, in chapter 5 we will demonstrate some of these ideas on the recovery of a protein-signaling network, the modeling of gene expression time-series, the recovery of the human skeleton from motion capture 3-D cloud data, the recovery/mapping of human poses from silhouettes and the discovery of collusion patterns within voting data from the annual Eurovision song contest.

3.1 Background

3.1.1 Probabilistic principal component analysis

Probabilistic principal component analysis (PPCA) decomposes the covariance of a multivariate random variable $\mathbf{y} \in \mathbb{R}^p$, into the sum of a low-rank term $\mathbf{W}\mathbf{W}^\top$ and a spherical noise term $\sigma^2\mathbf{I}$. The underlying probabilistic model assumes that each datum is Gaussian-distributed:

$$\mathbf{y} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}\right) \ , \tag{3.1}$$

where, without loss of generality, we center the datapoints (mean is zero) and $\mathbf{W} \in \mathbb{R}^{p \times q}$, with $q < p$, induces a reduced rank structure on the covariance. Thereby, the log-likelihood of the centered dataset $\mathbf{Y} \in \mathbb{R}^{n \times p}$ of n datapoints and p features or variables is:

$$p(\mathbf{Y}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_i \,|\, \mathbf{0}, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}\right) \ . \tag{3.2}$$

It was simultaneously and independently conjectured by Roweis [1998] and proven by Tipping & Bishop [1999] that this marginal likelihood, as a function of the loadings $\mathbf{W}$ and for a particular latent dimensionality $k = q$, is maximised when

$$\widehat{\mathbf{W}} = \mathbf{U}_q \mathbf{L}_q \mathbf{R}^\top \ , \tag{3.3}$$

where $\mathbf{U}_q$ is the eigenvector matrix with its columns being the q principal (orthonormal) eigenvectors of the sample-covariance matrix

$$\widehat{\mathbf{S}} \triangleq \tfrac{1}{n} \mathbf{Y}^\top \mathbf{Y} \ ,$$

ordered by the magnitudes of the corresponding eigenvalues. The $q \times q$ diagonal matrix $\mathbf{L}_q$ has elements

$$L_{q,ii} = \sqrt{\lambda_i - \sigma^2} \ , \tag{3.4}$$

with λ_i being the i-th largest eigenvalue of the sample-covariance matrix $\widehat{\mathbf{S}}$ and σ^2 being the noise variance, followed by an *arbitrary* orthogonal/rotation matrix $\mathbf{R}$. Note that the maximum-likelihood solution of the covariance in eq. (3.2), irrespective of q, is expressed in terms of the *singular value decomposition* (SVD, see appendix A.1) of $\widehat{\mathbf{W}}$, thus the maximum-likelihood solution is *rotation-invariant*, that is,

$$\widehat{\mathbf{W}}\widehat{\mathbf{W}}^\top = (\mathbf{U}_q \mathbf{L}_q \mathbf{R}^\top)(\mathbf{R} \mathbf{L}_q \mathbf{U}_q^\top) = \mathbf{U}_q \mathbf{L}_q^2 \mathbf{U}_q^\top$$

leads to the same positive semi-definite component in the covariance, for any rotation matrix $\mathbf{R}$. Intuitively, the matrix $\mathbf{W}$ spans the *principal subspace* or *latent space* within the data space, with respect to which, the latent subspace basis can have any relative (rigid) rotation that *does not affect* the covariances between the observed variables/features.

Generative low-rank models Underlying this model is an assumption that the data set is generated as

$$\mathbf{Y} = \mathbf{X}\mathbf{W}^\top + \mathbf{E} \ ,$$

where $\mathbf{X} \in \mathbb{R}^{n \times q}$ is the matrix of the low-dimensional latent representations $\mathbf{x} \in \mathbb{R}^q$ of the datapoints $\mathbf{y} \in \mathbb{R}^p$ and $\mathbf{E}$ is the matrix of noise variables,

$$E_{ij} \sim \mathcal{N}\left(0, \sigma^2\right) \ .$$

We diverge momentarily to note a tight connection to the problem of *multi-output* linear regression,

$$\mathbf{y}_i = \mathbf{W}\mathbf{x}_i + \boldsymbol{\epsilon} \ . \tag{3.5}$$

The main difference is that the *input* to the low rank system is now *unknown* (instead of the output and the system has *more outputs* than inputs, hence there is some *redundancy* in the system output.

Under this linear regression view, we can say that the "outputs" of the low-rank system are *conditionally independent* given the inputs. This redundancy *confounds* the underlying structure of the observed covariates. Loosely speaking:

> *Anything said by a set of highly agreeing variables can be equally expressed by fewer variables, up to a small error.*

See Figure 3.1 for a visualisation. This idea motivated the study of low-rank models like PCA [Jolliffe, 2002; Hotelling, 1933; see also Pearson, 1901 for historical purposes], *factor analysis* [Bartholomew *et al.*, 2011; Basilevsky, 1994] and *canonical correlation analysis* [Hotelling, 1936] (see section 4.2 for a review).

The combination of the linear mapping and Gaussian iid noise assumption gives the data likelihood:

$$p(\mathbf{Y} \mid \mathbf{X}, \mathbf{W}, \sigma^2) = \prod_i \mathcal{N}\left(\mathbf{y}_i \mid \mathbf{W}\mathbf{x}_i, \ \sigma^2\mathbf{I}\right) \ , \tag{3.6}$$

where i indexes the rows of $\mathbf{Y}$ and $\mathbf{X}$. Then, the marginal likelihood from eq. (3.2) is obtained by inducing a factorised Gaussian spherical prior[1] on (each row-vector

[1] We can use a $\mathcal{N}(\mathbf{0}, \mathbf{I})$ prior here for simplicity without loss of generality, since the functional form of $p(\mathbf{Y})$ remains unchanged for a general Gaussian prior.

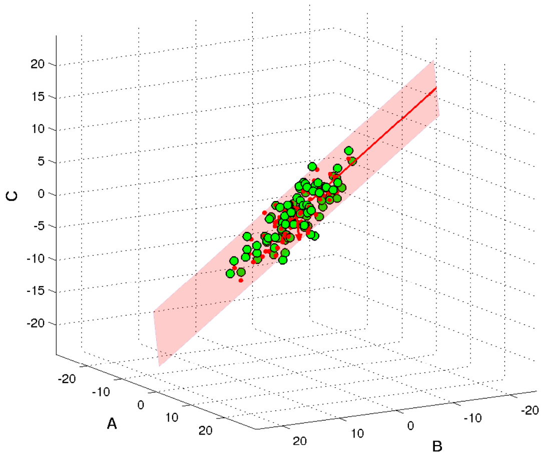


Figure 3.1: The redundancy in the observations (green dots) of A,B and C take the form of a linear manifold (red plane), spanned by the two principal eigenvectors of the sample-covariance (red arrows). The eigen-basis coordinates faithfully represent the original observations.

of) $\mathbf{X}$

$$p(\mathbf{X}) = \prod_i \mathcal{N}\left(\mathbf{x}_i \,|\, \mathbf{0}, \mathbf{I}\right) \tag{3.7}$$

and averaging over $\mathbf{X}$ with respect to its prior (see appendix A.1):

$$p(\mathbf{Y}) = \int \mathrm{d}\mathbf{X}\; p(\mathbf{Y} \,|\, \mathbf{X}, \mathbf{W}, \sigma^2)\; p(\mathbf{X})\;, \tag{3.8}$$

where, for the rest of the chapter, we suppress the parameters of the marginal distribution $p(\mathbf{Y} \,|\, \mathbf{W}, \sigma^2)$ to reduce cluttering. The graphical model of PPCA is illustrated in Figure 3.2(a): The interpretation of the marginal $p(\mathbf{Y})$ here is that in the system there exists a *fixed*[1] and *unknown* linear mapping $\mathbf{W}$ and each *sample* $\mathbf{y}_i$ is a contribution from a *single* latent point $\mathbf{x}_i$. The set of features of $\mathbf{x}_i$ are weighted differently (by a different *row* of $\mathbf{W}$) for each output or component of $\mathbf{y}_i$, see eq. (3.5).

[1] As in, non-random.

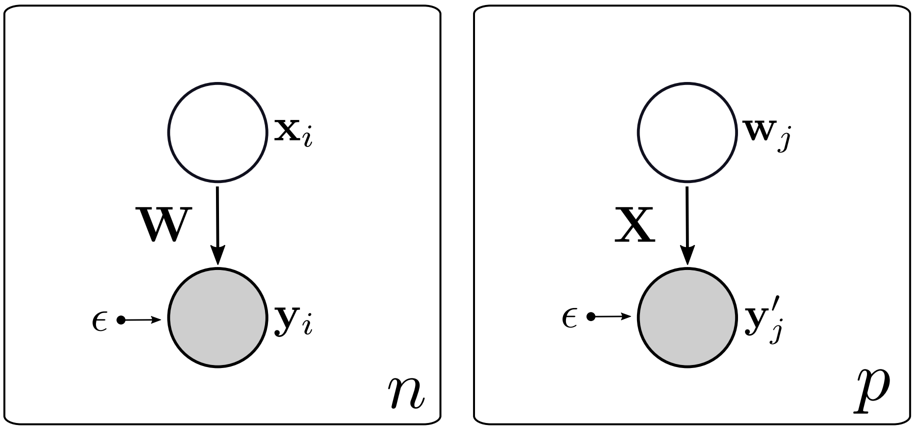


Figure 3.2: The shaded nodes distinguish the observed variables from the latent ones. (a) Graphical model of probabilistic PCA. The joint distribution factorises across the n datapoints (indexed by i) and the conditional likelihood is governed by the *mapping* $\mathbf{W}$. (b) Graphical model of dual PPCA. The joint distribution factorises across the p features (indexed by j) and the conditional likelihood is governed by the *latent coordinates* $\mathbf{X}$.

3.1.2 Dual PPCA

There is also an alternative interpretation of the marginal $p(\mathbf{Y})$, see Figure 3.2(b): In the system, there exists a *fixed*[1] set of latent points $\mathbf{X}$ and each *output* $\mathbf{y}'_j$ (column of $\mathbf{Y}$) is a contribution from *all* latent features (columns) in $\mathbf{X}$, such that

$$\mathbf{y}'_j = \mathbf{X}\mathbf{w}_j + \boldsymbol{\epsilon} \ ,$$

where the columns $\mathbf{x}_j$ are combined differently (by a different *row* $\mathbf{w}_j$ of $\mathbf{W}$) for each output $\mathbf{y}'_j$. This was the motivation for Lawrence [2005] when he showed that the PCA solution is also obtained for likelihoods of a *dual* form, recovered when we average over the loadings $\mathbf{W}$ with a (similarly) factorised Gaussian isotropic prior $p(\mathbf{W}) = \prod_j \mathcal{N}\left(\mathbf{w}_j \,|\, \mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{w}}\right)$, with diagonal $\boldsymbol{\Sigma}_{\mathbf{w}}$, instead of averaging over the latent points $\mathbf{X}$:

$$p(\mathbf{Y}) = \int \mathrm{d}\mathbf{W}\ p(\mathbf{Y}|\mathbf{X},\mathbf{W},\sigma^2)\ p(\mathbf{W}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}'_j \,|\, \mathbf{0}, \mathbf{X}\boldsymbol{\Sigma}_{\mathbf{w}}\mathbf{X}^{\top} + \sigma^2\mathbf{I}\right) \ , \tag{3.9}$$

where j indexes the *columns* of $\mathbf{Y}$ and the parameters in the marginal likelihood are suppressed as before.

Identifiability Because of the diagonal prior covariance $\boldsymbol{\Sigma}_{\mathbf{w}}$, there is an indeterminacy between $\mathbf{X}$ and $\boldsymbol{\Sigma}_{\mathbf{w}}$ in the maximum-likelihood solution, meaning that $\mathbf{X}\boldsymbol{\Sigma}_{\mathbf{w}}^{1/2}$ would also be a solution and there is no way to distinguish between the two. A principled way to work around this is to assume that $\mathbf{X}^\top\mathbf{X} = \mathbf{I}$ which is equivalent to seeking a MAP solution for the latent variables under a spherical Gaussian prior on $\mathbf{x}$; we stick to this approach for the sake of simplicity. The marginal likelihood is now parameterised by the latent points $\mathbf{X}$ instead of the loadings $\mathbf{W}$ and the factorisation now implies conditional independences across the p *features*, as opposed to the n datapoints, meaning that the *covariances* are expressed between *datapoints* and not features.

Dual interpretation This *dual* formulation of PCA[1] is also known as *principal coordinate analysis* as it solves for the the *latent coordinates* instead of the principal subspace basis and the maximum-likelihood solution is now:

$$\widehat{\mathbf{X}} = \mathbf{U}_q'\mathbf{L}_q\mathbf{R}^\top ,$$

where $\mathbf{L}_q$ and $\mathbf{R}$ is defined as in eq. (3.3) and the columns of $\mathbf{U}_q'$ are the first q left-singular vectors of $\mathbf{Y}$, or equivalently, the q principal eigenvectors of the sample-inner product (feature covariance) matrix $\mathbf{Y}\mathbf{Y}^\top$ (see **SVD**, appendix A.3). The rotation $\mathbf{R}$ introduces a second kind of indeterminacy, but as we discuss in a later section, the rotation is not important (set as the identity) since we most often care about the relative positions of the latent variables.

Connection to the Gaussian process The underlying model in eq. (3.9) is in fact a product of independent *Gaussian processes* with *linear* covariance functions, see section 2.2.1 and [Rasmussen & Williams, 2006]. In this form, the

[1] This contrasts the typical primal form. The name refers to the *duality* between the sample-space (row-space) and the feature-space (column-space) of a typical design matrix $\mathbf{Y}$ with its rows as samples.

generalisation to a non-linear mapping from the latent space $\mathcal{X}$ to the observed space $\mathcal{Y}$ now seems almost straightforward:

$$p(\mathbf{Y}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}'_j \,|\, \mathbf{0}, \mathbf{K} + \sigma^2\mathbf{I}\right) \ ,$$

where we have replaced the (bi-)linear covariance function, defined by the inner product between samples $\mathbf{x}^\top\mathbf{x}'$, with a non-linear covariance function $k(\mathbf{x}, \mathbf{x}')$, see section 2.2.2 for an example. The resulting model is known as the *Gaussian process latent variable model* (GPLVM) [Lawrence, 2005], most notably used for non-linear dimensionality reduction [Lawrence, 2004], with Bayesian extensions thereof [Damianou *et al.*, 2011; Titsias & Lawrence, 2010].

3.2 Low-rank plus positive definite covariance

Both primal and dual interpretations involve maximising Gaussian likelihoods of a similar covariance structure, namely, that of a *low-rank plus a spherical noise* term. In many parts of this chapter we motivate the framework while focusing on the dual case,

$$\mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I} \ ,$$

without loss of generality for the primal case. Where possible, we give the primal cases of the equations as well. The focus of this chapter is a more general form of the above covariance structure given by

$$\text{dual:} \quad \mathbf{X}\mathbf{X}^\top + \boldsymbol{\Sigma} \tag{3.10}$$

$$\text{primal:} \quad \mathbf{W}\mathbf{W}^\top + \boldsymbol{\Sigma} \ , \tag{3.11}$$

where $\boldsymbol{\Sigma}$ is a *positive definite* matrix. We are motivated by scenarios where the data variance is already *partly explained* by the covariance term $\boldsymbol{\Sigma}$ and we wish to study the components of the *residual variance*. We show that our ideas can be applied for both primal and dual representations and the representation of choice depends on the information that we wish to encode in $\boldsymbol{\Sigma}$.

3.2.1 Motivating examples

Consider the general functional form of a *linear mixed-effects* model [Pinheiro & Bates, 2000] with two factors $\mathbf{X}, \mathbf{Z}$ and noise $\mathbf{E}$:

$$\mathbf{Y} = \mathbf{X}\mathbf{W}^\top + \mathbf{Z}\mathbf{V}^\top + \mathbf{E} \,, \tag{3.12}$$

where $\mathbf{Z}$ is a matrix of known covariates (*fixed* effects) with some predictive power for $\mathbf{Y}$, and $\mathbf{X}$ is a matrix of latent variables (*random* effects), see also [Fusi *et al.*, 2012] for an application with Gaussian processes on genome-wide association studies. This linear mixed-effects model will serve as a conceptual reference point for the rest of this chapter and it is illustrated in Figure 3.3(a). In this section, we mention a few specific forms that eq. (3.12) can take for various applications including one of visualisation and computational biology.

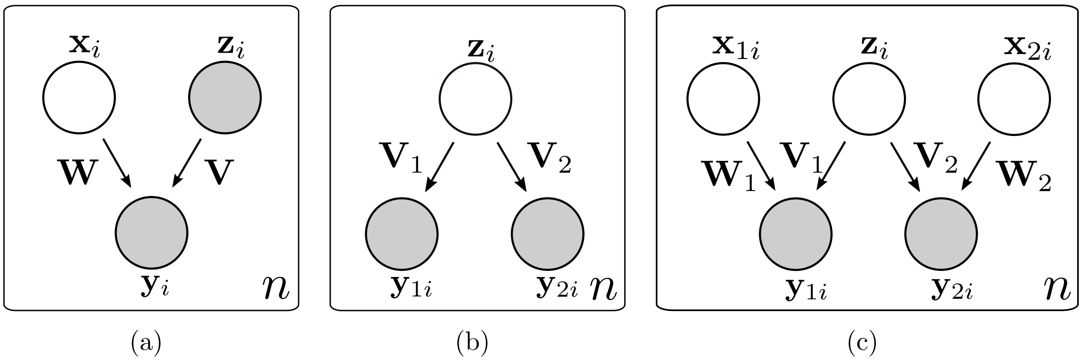


Figure 3.3: **(a)** A linear mixed-effects model. *Fixed effects* $\mathbf{z}$ partially explain the variance in the observation $\mathbf{y}$ through the mapping defined by $\mathbf{V}$. The *residual variance* is then explained by *random effects* $\mathbf{x}$ up to noise. **(b)** Probabilistic CCA model. Observations $\mathbf{y}_1$ and $\mathbf{y}_2$ share the latent variable $\mathbf{z}$, thus the variance in the joint data is explained solely by $\mathbf{z}$ up to noise. In other words, the model assumes no structure within $\mathbf{y}_1$ or $\mathbf{y}_2$ but only *between* their two covariate sets. **(c)** Linear *multi-view learning* model, also known as *inter-battery factor analysis* [IBFA, Ek *et al.*, 2008; Klami & Kaski, 2006; Tucker, 1958] , where observations $\mathbf{y}_1$ and $\mathbf{y}_2$ *share* the latent variable $\mathbf{z}$ but also have *private* latent variables $\mathbf{x}_1$ and $\mathbf{x}_2$ respectively.

Akin to averaging over the loadings $\mathbf{W}$ as we did in eq. (3.9) (or over the factors $\mathbf{X}$ as in eq. (3.8)), in the mixed-model case we can also average over the loadings $\mathbf{V}$ (or factors $\mathbf{Z}$ for the primal) to recover the likelihood:

$$\begin{aligned}\textbf{(dual)} \qquad p(\mathbf{Y}) &= \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}'_j \,|\, \mathbf{0}, \mathbf{X}\mathbf{X}^\top + \boldsymbol{\Sigma}\right) \\ \textbf{(primal)} \qquad p(\mathbf{Y}) &= \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_i \,|\, \mathbf{0}, \mathbf{W}\mathbf{W}^\top + \boldsymbol{\Sigma}\right) \ ,\end{aligned} \tag{3.13}$$

where the positive definite matrix $\boldsymbol{\Sigma}$ assumes the role of the *explained variance*:

$$\begin{aligned}\textbf{(dual)} \qquad \boldsymbol{\Sigma} &= \mathbf{Z}\mathbf{Z}^\top + \sigma^2\mathbf{I} \\ \textbf{(primal)} \qquad \boldsymbol{\Sigma} &= \mathbf{V}\mathbf{V}^\top + \sigma^2\mathbf{I} \ .\end{aligned}$$

For instance, the representation of $\mathbf{Y}$ in eq. (3.12) can manifest as:

(a) a set of protein activation signals under various external stimuli (which make the data heterogeneous). In this *primal* scenario, $\mathbf{V}$ is the identity matrix and there are as many effects (columns) in $\mathbf{Z}$ as there are covariates in $\mathbf{Y}$. The factors in $\mathbf{Z}$ are special in that they share a *sparse* network of *conditional dependencies*. Sparse dependencies are interesting in terms of learning parsimonious models but in realistic scenarios the sparsity can be *confounded* by the heterogeneous experimental conditions (the various stimuli) under which $\mathbf{Y}$ is generated. We encode these *confounders* as the factors (columns) of $\mathbf{X}$. Intuitively, if the confounders are fewer than our observed covariates and the confounders somehow affect the observed space in a linear fashion $\mathbf{X}\mathbf{W}^\top$, then the variance explained solely by $\mathbf{X}$ is a *low-rank* term in the marginal covariance, see eq. (3.13, primal). Another way to argue about this is by seeing how the nominal values of our activation signals are *forced* to diverge from their otherwise true values: Because there are always fewer confounders than covariates, there is *redundancy* in the way the confounders express in the observed space and, consequently, a low-rank structure in the covariance of our measurements.

Returning to the sparse dependencies, we are led to parameterise the explained covariance term as:

$$\mathbf{\Sigma}_{GMRF} = \mathbf{\Lambda}^{-1} \ , \tag{3.14}$$

where $\mathbf{\Lambda}$ is sparse, thus recovering a *low-rank plus sparse-inverse* parameterisation of the covariance in eq. (3.13). A sparse precision inscribes a sparsely connected *Gaussian Markov random field* (GMRF) or a *Gaussian graphical model* of the factors in $\mathbf{Z}$, such that each row $\mathbf{z}_i$ is distributed from $\mathcal{N}\left(\mathbf{0}, \mathbf{\Lambda}^{-1}\right)$, where the precision matrix $\mathbf{\Lambda}$ is sparse [Lauritzen, 1996];

(b) a set of n gene expression profiles as rows, where each profile concatenates two time-series of p_1 timepoints sampled under control conditions plus p_2 timepoints sampled under test conditions. In this *dual* scenario, the instantiation

$$\mathbf{\Sigma}_{Gram} = \mathbf{K} + \sigma^2 \mathbf{I} \ ,$$

for a general *Gram* matrix $\mathbf{K}$, expresses temporal correlations in a time-series dataset, with $K_{ij} = k(\mathbf{z}_i, \mathbf{z}_j)$ for some covariance function $k : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$. This approach gets close to the common practice of explicitly subtracting the result of a simpler model from the data and then analyzing the residual separately;

(c) a set of n patients' gene expression measurements of p genes, with each row of $\mathbf{Z}$ being the set of genotypes[1] of each patient and each row of $\mathbf{X}$ being some unobserved environmental effects (confounders), see [Fusi *et al.*, 2012].

In each of these cases, the benefit of analysing the components of the *residual* variance $\mathbf{X}\mathbf{X}^\top$ (primal: $\mathbf{W}\mathbf{W}^\top$), given the *explained* variance $\mathbf{\Sigma}$ or some estimate thereof, is twofold: on the one hand, learning about the confounders and potentially correcting our data for their effects and on the other hand, learning $\mathbf{\Sigma}$ when we restrict it to a specific type of structure (e.g. sparse-inverse). This raises the following two questions:

1. *Given* $\mathbf{\Sigma}$*, how can we solve for* $\mathbf{X}$ *(respectively* $\mathbf{W}$*)?*

And more importantly:

[1] For example, SNP (single-nucleotide polymorphism) data.

2. *For what forms of* $\mathbf{\Sigma}$ *can we formulate useful new algorithms for machine learning?*.

We refer to the forms of $\mathbf{\Sigma}$ as *instantiations* for the rest of this chapter and denote them by a subscript.

3.2.2 Proposed approach

First, the key theoretical result of this chapter in Section 3.3.1 shows that the maximum-likelihood solution for $\mathbf{X}$ (primal: $\mathbf{W}$) is simply based on a *generalised eigenvalue problem* (GEP) of the sample-covariance matrix and explained covariance $\mathbf{\Sigma}$. Hence, the low-rank term $\mathbf{X}\mathbf{X}^\top$ of the marginal covariance can be optimized for an arbitrary fixed positive definite $\mathbf{\Sigma}$. We call this data analysis approach *residual component analysis* (RCA). De Bie *et al.* [2005] present a nice review on a range of GEPs in the machine learning literature.

Secondly, from a unification viewpoint the RCA approach is interesting as it connects a few classical methods and their probabilistic counterparts in the literature and also gives rise to a range of new algorithms suited for the aforementioned scenarios. For instance, for scenario (a) we propose an EM/RCA hybrid algorithm in section 5.1.1 for estimating both the low-rank and sparse-inverse terms. For scenario (c) we present a pure RCA treatment in Section 5.2: the residual basis of interest is found with a single estimate via the GEP solution. The focus of chapter 5 and is on demonstrating the effectiveness of the algorithms on a variety of datasets and application domains.

3.2.3 Background

GLASSO The *low-rank plus inverse-sparse* parameterisation, by eqs. (3.13-3.14), extends the *Graphical Lasso* (GLASSO) algorithm [Banerjee *et al.*, 2008; Friedman *et al.*, 2008]. GLASSO is a MAP approach to maximising the Gaussian likelihood, as a function of the covariance, with an l_1-*penalty* (sparsity-promoting) term on the precision matrix $\mathbf{\Lambda}$:

$$\max_{\mathbf{\Lambda}} \left\{ \ln|\mathbf{\Lambda}| - \operatorname{tr}(\widehat{\mathbf{S}}\mathbf{\Lambda}) - \lambda||\mathbf{\Lambda}||_1 \right\} \ , \tag{3.15}$$

where $\widehat{\mathbf{S}}$ is the sample-covariance matrix. Due to the L1 restriction on the solution norm, the stationarity conditions no longer have a closed form. Nonetheless, the problem is still convex and a global solution is found efficiently through the iterative use of *least angle regression* [Hastie *et al.*, 2009]. Sparse-inverse structures capture relations between variables that are not well characterized by low-rank forms. As such, the combination of sparse-inverse and low-rank can be a powerful one with applications in computational biology and visualisation, as we demonstrate in chapter 5. We also point to the work of Stegle *et al.* [2011] for a different approach based on a multiplicative — Kronecker product — structure in the covariance.

PPCA We also note a few more connections to well-studied algorithms for linear dimensionality reduction. The obvious connection to PPCA is recovered by

$$\mathbf{\Sigma}_{PCA} = \sigma^2 \mathbf{I} \ .$$

Bi-directed graphs If the covariance term $\mathbf{\Sigma}$ is assumed to be *sparse* (as opposed to sparse-*inverse*), then this relates to the problem of structure learning for Gaussian bi-directed graphs [Silva, 2011]. Such graphs encode constraints of marginal independence and are of interest due to being closed under marginalisation (that is, the graph retains its set of independencies over the remaining variables), [Richardson & Spirtes, 2002].

PCCA More interestingly, *probabilistic canonical correlation analysis* (PCCA) [Bach & Jordan, 2002, 2005] is recovered by

$$\mathbf{\Sigma}_{CCA} = \begin{bmatrix} \mathbf{Y}_1^\top \mathbf{Y}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_2^\top \mathbf{Y}_2 \end{bmatrix} \ , \quad \text{for the concatenation} \quad \mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \end{bmatrix} .$$

We prove this non-trivial statement in section 4.3.

Inter-battery factor analysis On a similar note, if

$$\mathbf{Y} = \begin{bmatrix} \mathbf{X}_1\mathbf{W}_1^\top + \mathbf{Z}\mathbf{V}_1^\top + \mathbf{E}_1 \\ \mathbf{X}_2\mathbf{W}_2^\top + \mathbf{Z}\mathbf{V}_2^\top + \mathbf{E}_2 \end{bmatrix},$$

then the partitions of $\mathbf{Y}$ have their own associated private latent spaces of $\mathbf{X}_1$ and $\mathbf{X}_2$, in addition to the standard shared latent space of $\mathbf{Z}$ found in CCA, see Figure 3.3(c). This is in fact a special case of the *multi-view learning* model of Ek *et al.* [2008]; the linear case was more closely studied by Klami & Kaski [2006, 2008] and is known as *extended probabilistic-CCA*. In the statistics literature the model is known as *inter-battery factor analysis* (IBFA) [Browne, 1979; Tucker, 1958]. To train this type of model, an *iterative* treatment of RCA can be formulated; we give an outline here: on step one, solve for the weights $\mathbf{V}$ of the *shared components* by setting the explained covariance term as

$$\mathbf{\Sigma}_{IBFA} = \begin{bmatrix} \mathbf{W}_1\mathbf{W}_1^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_2\mathbf{W}_2^\top \end{bmatrix} + \sigma^2\mathbf{I}$$

in the GEP of the concatenated data sample-covariance matrix. On step two, for each of $d \in \{1, 2\}$ views solve for the weights $\mathbf{W}_d$ of the view-specific components by setting

$$\mathbf{\Sigma}_{IBFA} = \mathbf{V}_d\mathbf{V}_d^\top + \sigma^2\mathbf{I}$$

in the GEP of the sample-covariance associated with $\mathbf{Y}_k$. This *iterative-RCA* algorithm is reminiscent of the expectation-maximization (EM) algorithm for optimising extended-PCCA, as both approaches maximise the likelihood by fitting components into the residual. We provide more details of *iterative-RCA* algorithm in section 4.6.

Coloured noise models Lastly, we mention a link to existing work on *coloured noise* models from the signal processing literature, that is, linear models that assume a full noise matrix in the marginal covariance [Chen & Wang, 2006; Hu &

Loizou, 2003]. Such models mitigate noise effects by performing *oblique* (non-orthogonal) projections of the data *onto* the signal subspace but *along the direction of* the noise subspace [Behrens & Scharf, 1994]. As we discuss in section 3.3.3, oblique projections have an important role in interpreting the proof of the RCA theorem as well as providing a *geometric interpretation* of RCA as a data analysis tool. Ultimately, it turns out that the GEP of RCA is strongly tied to an oblique projection as it estimates either the oblique-projected data (dual) or the projection basis (primal), but the theorem is novel as it introduces a probabilistic interpretation of the recovered oblique projector subspace in the same way that PPCA enriched classical PCA, for both primal and dual representations.

3.3 Maximum-likelihood residual component analysis

We show the main theoretical results on the dual case, without loss of generality for the primal case.

3.3.1 RCA theorem

Dual case theorem

For a positive-definite $\mathbf{\Sigma}$ with a spectral radius at most as large as that of $\frac{1}{p}\mathbf{Y}\mathbf{Y}^\top$, the maximum-likelihood estimate of the parameters $\mathbf{X}$ of the marginal $p(\mathbf{Y}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}'_j \,|\, \mathbf{0}, \mathbf{X}\mathbf{X}^\top + \mathbf{\Sigma}\right)$ is

$$\widehat{\mathbf{X}} = \mathbf{\Sigma}\mathbf{S}(\mathbf{D}-\mathbf{I})^{1/2} \ , \tag{3.16}$$

where $\mathbf{S}$ is the solution to the GEP,

$$\tfrac{1}{p}\mathbf{Y}\mathbf{Y}^\top\mathbf{S} = \mathbf{\Sigma}\mathbf{S}\mathbf{D} \ , \tag{3.17}$$

and its columns are the generalised eigenvectors (of $\frac{1}{p}\mathbf{Y}\mathbf{Y}^\top$ and $\mathbf{\Sigma}$) and

$\mathbf{D}$ *is the diagonal matrix of the corresponding generalised eigenvalues.*

Proof The log-marginal likelihood $\ln p(\mathbf{Y})$, as a function of the latent variables $\mathbf{X}$, is:

$$L(\mathbf{X}) = -\frac{p}{2}\ln|\mathbf{K}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{Y}\mathbf{Y}^\top\mathbf{K}^{-1}\right) - \frac{np}{2}\ln(2\pi)\ ,$$

where $\mathbf{K} \triangleq \mathbf{X}\mathbf{X}^\top + \mathbf{\Sigma}$. Since $\mathbf{\Sigma}$ is positive-definite, we can consider its eigendecomposition:

$$\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top, \tag{3.18}$$

where $\mathbf{U}^\top\mathbf{U} = \mathbf{U}\mathbf{U}^\top = \mathbf{I}$ and $\mathbf{\Lambda}$ is diagonal (see appendix A.3).

We proceed by rotating the marginal covariance $\mathbf{K}$ from the data-space basis to the eigen-basis $\mathbf{U}$ and scaling by the eigenvalues $\mathbf{\Lambda}$:

$$\begin{aligned}\widetilde{\mathbf{K}} &\triangleq \mathbf{\Lambda}^{-1/2}\mathbf{U}^\top\mathbf{K}\mathbf{U}\mathbf{\Lambda}^{-1/2}\\ &= \mathbf{\Lambda}^{-1/2}\mathbf{U}^\top\left(\mathbf{X}\mathbf{X}^\top + \mathbf{\Sigma}\right)\mathbf{U}\mathbf{\Lambda}^{-1/2}\\ &= \left(\mathbf{\Lambda}^{-1/2}\mathbf{U}^\top\mathbf{X}\right)\left(\mathbf{X}^\top\mathbf{U}\mathbf{\Lambda}^{-1/2}\right) + \mathbf{I}\\ &= \widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^\top + \mathbf{I}\ ,\end{aligned} \tag{3.19}$$

where we have defined the rotated and scaled latent points

$$\widetilde{\mathbf{X}} \triangleq \mathbf{\Lambda}^{-1/2}\mathbf{U}^\top\mathbf{X}\ . \tag{3.20}$$

Therefore, from the inverse-transformation of $\mathbf{K}$ in eq. (3.19) we get the determinant and trace

$$\begin{aligned}|\mathbf{K}| &= |\widetilde{\mathbf{K}}||\mathbf{\Lambda}|\\ \mathrm{tr}\left(\mathbf{Y}\mathbf{Y}^\top\mathbf{K}^{-1}\right) &= \mathrm{tr}\left(\mathbf{\Lambda}^{-1/2}\mathbf{U}^\top\mathbf{Y}\mathbf{Y}^\top\mathbf{U}\mathbf{\Lambda}^{-1/2}\widetilde{\mathbf{K}}^{-1}\right)\\ &= \mathrm{tr}\left(\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^\top\widetilde{\mathbf{K}}^{-1}\right)\ ,\end{aligned}$$

where, in a similar manner, we have transformed the data

$$\widetilde{\mathbf{Y}} \triangleq \mathbf{\Lambda}^{-1/2}\mathbf{U}^{\top}\mathbf{Y} \ . \tag{3.21}$$

Now we are in position to re-parameterise the log-marginal likelihood as a function of the *transformed* variables $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$:

$$L(\widetilde{\mathbf{X}}) = -\frac{p}{2}\ln\left(|\widetilde{\mathbf{K}}||\mathbf{\Lambda}|\right) - \frac{1}{2}\mathrm{tr}\left(\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^{\top}\widetilde{\mathbf{K}}^{-1}\right) - \frac{np}{2}\ln(2\pi) \ .$$

We know how to maximize this new form of the log-likelihood, by following a route similar to the proof of Tipping & Bishop [1999]: Taking the gradient with respect to the new parameters $\widetilde{\mathbf{X}}$ (see **Matrix derivatives**, appendix A.2),

$$\frac{\partial L}{\partial \widetilde{\mathbf{X}}} = \widetilde{\mathbf{K}}^{-1}\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^{\top}\widetilde{\mathbf{K}}^{-1}\widetilde{\mathbf{X}} - p\,\widetilde{\mathbf{K}}^{-1}\widetilde{\mathbf{X}} \ ,$$

gives the stationary point

$$\widetilde{\mathbf{X}} = \frac{1}{p}\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^{\top}\widetilde{\mathbf{K}}^{-1}\widetilde{\mathbf{X}} \ . \tag{3.22}$$

Next, we replace $\widetilde{\mathbf{X}}$ in eq. (3.22) with its SVD,

$$\widetilde{\mathbf{X}} = \widetilde{\mathbf{V}}\mathbf{L}\mathbf{R}^{\top} \ , \tag{3.23}$$

which gives

$$\widetilde{\mathbf{V}}\mathbf{L}\mathbf{R}^{\top} = \frac{1}{p}\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^{\top}\left(\widetilde{\mathbf{V}}\mathbf{L}^{2}\widetilde{\mathbf{V}}^{\top} + \mathbf{I}\right)^{-1}\widetilde{\mathbf{V}}\mathbf{L}\mathbf{R}^{\top} \ .$$

By applying[1] the *Woodbury matrix identity* (see appendix A.3.3) and simplifying, we see that maximisation relies on the *regular eigenvalue problem*:

$$\frac{1}{p}\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^{\top}\widetilde{\mathbf{V}} = \widetilde{\mathbf{V}}\mathbf{D}, \quad \text{where} \quad \mathbf{D} \triangleq \mathbf{L}^{2} + \mathbf{I} \ . \tag{3.24}$$

Now, we focus on relating the stationary point of $\widetilde{\mathbf{X}}$ to that of $\mathbf{X}$. First, we

[1] Here, it is assumed that $\mathbf{L}$ is square and diagonal and $\widetilde{\mathbf{V}}$ is rectangular. If we start this step with an orthonormal $\widetilde{\mathbf{V}}$ but rectangular $\mathbf{L}$ then we end up with $\widetilde{\mathbf{V}}_q$ in eq. (3.24), keeping only the first q columns.

express the eigenvalue problem of eq. (3.24) in terms of $\mathbf{YY}^\top$. To do that, we use the definition of $\widetilde{\mathbf{X}}$ from eq.(3.20) to obtain the SVD of $\mathbf{X}$:

$$\mathbf{X} = \left(\mathbf{U}\mathbf{\Lambda}^{1/2}\widetilde{\mathbf{V}}\right)\mathbf{LR}^\top = \mathbf{VLR}^\top \ , \tag{3.25}$$

where $\mathbf{V} \triangleq \mathbf{U}\mathbf{\Lambda}^{1/2}\widetilde{\mathbf{V}}$ are the left-singular vectors of $\mathbf{X}$. This makes explicit the relationship between the row-spaces of $\mathbf{X}$ and $\widetilde{\mathbf{X}}$. Then, we substitute $\widetilde{\mathbf{Y}}$ and $\widetilde{\mathbf{V}}$ with their definitions in eq.(3.24) and use the inverse of $\mathbf{\Sigma}$ from eq.(3.18) to recover the equivalent eigenvalue problem:

$$\begin{aligned}\tfrac{1}{p}\left(\mathbf{\Lambda}^{-1/2}\mathbf{U}^\top\mathbf{Y}\right)\left(\mathbf{Y}^\top\mathbf{U}\mathbf{\Lambda}^{-1/2}\right)\left(\mathbf{\Lambda}^{-1/2}\mathbf{U}^\top\mathbf{V}\right) &= \left(\mathbf{\Lambda}^{-1/2}\mathbf{U}^\top\mathbf{V}\right)\mathbf{D}\\ \tfrac{1}{p}\mathbf{YY}^\top\left(\mathbf{U}\mathbf{\Lambda}^{-1}\mathbf{U}^\top\right)\mathbf{V} &= \mathbf{VD}\\ \tfrac{1}{p}\mathbf{YY}^\top\mathbf{\Sigma}^{-1}\mathbf{V} &= \mathbf{VD} \ .\end{aligned}$$

To conclude the proof, we define $\mathbf{S} \triangleq \mathbf{\Sigma}^{-1}\mathbf{V}$ to recover the desired *symmetric* form of the GEP:

$$\frac{1}{p}\mathbf{YY}^\top\mathbf{S} = \mathbf{\Sigma}\mathbf{SD} \ .$$

Based on the SVD of $\mathbf{X}$ in eq. (3.25), now we can recover $\mathbf{X}$ up to rotation $\mathbf{R}$ — which for simplicity is normally set to the identity — and rank q via the first q generalised eigenvectors of $\mathbf{YY}^\top$:

$$\mathbf{X}_q \quad = \quad \mathbf{V}_q\mathbf{L}_q \quad = \quad \mathbf{\Sigma}\mathbf{S}_q\mathbf{L}_q \quad = \quad \mathbf{\Sigma}\mathbf{S}_q(\mathbf{D}_q - \mathbf{I})^{1/2} \ . \qquad \square$$

Primal case theorem The algebraic symmetry between the primal and dual formulations of the marginal likelihood, eq. (3.13), allows us to easily extend the theorem to the primal case. Specifically,

> *the maximum-likelihood solution of the parameters* $\mathbf{W}$ *of the marginal* $p(\mathbf{Y}) = \prod_{i=1}^{n}\mathcal{N}\left(\mathbf{y}_j \,|\, \mathbf{0}, \mathbf{WW}^\top + \mathbf{\Sigma}\right)$, *has the same functional form,*
>
> $$\widehat{\mathbf{W}} = \mathbf{\Sigma}\mathbf{S}(\mathbf{D} - \mathbf{I})^{1/2} \ , \tag{3.26}$$

where the columns of $\mathbf{S}$ *are the generalised eigenvectors of the GEP:*

$$\frac{1}{n}\mathbf{Y}^{\top}\mathbf{Y}\mathbf{S} = \mathbf{\Sigma}\mathbf{S}\mathbf{D} \ . \tag{3.27}$$

Commentary To summarise the proof strategy, we start with the marginal $p(\mathbf{Y})$ with a *low-rank plus full noise* covariance structure and we re-express it in terms of a *low-rank plus spherical noise* covariance, by essentially rotating and scaling the data-space as per the spectral decomposition of the explained covariance term $\mathbf{\Sigma}$. This results in the transformed marginal[1] $p(\widetilde{\mathbf{Y}})$ with a *low-rank plus diagonal noise* covariance structure, as in PPCA. At this point, we can use the main result of Tipping & Bishop [1999] to compute the maximum-likelihood estimate of the parameters of the new distribution. Finally, using the estimates and their relation to the original parameters, we can solve for the parameters of the original distribution.

Aside from the generality of $\mathbf{\Sigma}$, we note a subtle difference from the PPCA solution for $\mathbf{W}$ in eq. (3.3, p. 39): Whereas PPCA in eq. (3.4) explicitly subtracts the noise variance from the q retained principal eigenvalues, RCA in eq. (3.19) implicitly incorporates any noise terms into $\mathbf{\Sigma}$ and *standardises* them when it projects the total covariance onto the eigen-basis of $\mathbf{\Sigma}$. Thus we get a reduction of unity from the retained generalised eigenvalues in eq. (3.16). As we discuss in more detail in section 4.3, for $\mathbf{\Sigma} = \mathbf{I}$ the PPCA and RCA solutions are the same.

3.3.2 Posterior expectation as an oblique projection

We know how to learn the mapping from the latent to the observed space and now we wish to infer the distribution over the latent variables; we use the primal picture for this, see Figure 3.3(a), p. 45.

Specifically, we wish to infer the posterior mean and covariance of $p(\mathbf{x}|\mathbf{y})$: By Bayes' theorem and conjugacy (both the likelihood $p(\mathbf{y}|\mathbf{x})$ and prior $p(\mathbf{x})$ are

[1] This is a linear transformation of the distribution domain, so the mode is preserved after the transformation.

Gaussian), the posterior distribution $p(\mathbf{x}|\mathbf{y})$ is also Gaussian. Hence, computing the posterior relies on completing the square in the exponent of the joint distribution:

$$p(\mathbf{x}\,|\,\mathbf{y}) \;\propto\; p(\mathbf{y}\,|\,\mathbf{x})\; p(\mathbf{x}) \;=\; \mathcal{N}\left(\mathbf{y}\,|\,\mathbf{W}\mathbf{x}, \boldsymbol{\Sigma}\right)\; \mathcal{N}\left(\mathbf{x}\,|\,\mathbf{0}, \mathbf{I}\right) \;,$$

where $\mathbf{y}$ is a centered datapoint and in the likelihood we have averaged over the fixed effects $\mathbf{z}$ so the explained covariance becomes $\boldsymbol{\Sigma} = \mathbf{V}\mathbf{V}^\top + \sigma^2\mathbf{I}$. Isolating the quadratic and linear terms in $\mathbf{x}$ in the exponent,

$$\begin{aligned} \ln p(\mathbf{x}|\mathbf{y}) \;&\propto\; -(p+1)\ln(2\pi) \;-\; |\boldsymbol{\Sigma}| \;-\; (\mathbf{y}-\mathbf{W}\mathbf{x})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\mathbf{W}\mathbf{x}) \;-\; \mathbf{x}^\top\mathbf{x} \\ &=\; C \;-\; \mathbf{y}^\top\boldsymbol{\Sigma}^{-1}\mathbf{y} \;+\; 2\mathbf{x}^\top\mathbf{W}^\top\boldsymbol{\Sigma}^{-1}\mathbf{y} \;-\; \mathbf{x}^\top(\mathbf{W}^\top\boldsymbol{\Sigma}^{-1}\mathbf{W}+\mathbf{I})\mathbf{x} \;, \end{aligned}$$

gives the covariance and mean of the posterior distribution $p(\mathbf{x}|\mathbf{y})$:

$$\begin{aligned} \boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{y}} &= (\mathbf{W}^\top\boldsymbol{\Sigma}^{-1}\mathbf{W}+\mathbf{I})^{-1} \\ \mathbb{E}\left[\mathbf{x}|\mathbf{y}\right] &= \boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{y}}\mathbf{W}^\top\boldsymbol{\Sigma}^{-1}\mathbf{y} \;. \end{aligned} \tag{3.28}$$

Similarly in the dual picture, Figure 3.2(b) (recall that $\boldsymbol{\Sigma} = \mathbf{Z}\mathbf{Z}^\top + \sigma^2\mathbf{I}$), we learn the latent coordinates $\mathbf{X}$ via RCA and then we can infer the posterior over a loadings vector $\mathbf{w}$ conditioned on a particular output $\mathbf{y}'$:

$$\begin{aligned} \boldsymbol{\Sigma}_{\mathbf{w}|\mathbf{y}'} &= (\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{X}+\mathbf{I})^{-1} \\ \mathbb{E}\left[\mathbf{w}|\mathbf{y}'\right] &= \boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{y}'}\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{y}' \;. \end{aligned}$$

The take-away message of this section is that when $\boldsymbol{\Sigma} = \mathbf{I}$ (reduced to PCA), the posterior expectation can be seen as the coordinates part[1] of an orthogonal projection on the column space of $\mathbf{W}$ with a bias towards zero due to the Gaussian prior. Similarly, what is reflected from the functional form of the posterior expectation in eq. (3.28) and its dual counterpart is the coordinates part of a *biased oblique projection*. This is illustrated on a toy example in the next section but more details follow.

[1] One way the projection formula can be broken down is [*Basis of subspace in* $\mathbb{R}^p$ *as the coordinate system*] × [*Coordinates*].

Explained covariance term as a null-steering operator

The following point is important, so we reproduce the primal form of the *biased oblique projector*:

$$\mathbf{W}(\mathbf{W}^\top\mathbf{\Sigma}^{-1}\mathbf{W}+\mathbf{I})^{-1}\mathbf{W}^\top\mathbf{\Sigma}^{-1} \tag{3.29}$$

which is similar to eq. (A.20) in appendix A.3.2, where we review the relevant properties of oblique projectors. Clearly, we are *stretching* the definition of a projector here, for two reasons. The first being that — strictly speaking — a projector must be *idempotent* (equal to its square) whereas eq. (3.29) is *biased* (due to the additive $\mathbf{I}$). Additionally, by terminology introduced in the same appendix, $\mathbf{\Sigma}^{-1}$ plays the role of a *null-steering operator*, that is, an orthogonal projector that nulls everything in the subspace spanned by the fixed-covariates: any observation in $\mathbb{R}^p$ is first applied with this null-steering operator that *orthogonally projects* onto the *orthogonal complement* of the *null-space* of the oblique projector. This is the null-space that governs the *directions* along which an oblique projection occurs and, with $\mathbf{\Sigma}^{-1}$ as the operator, the projecting directions are the same as the *principal components* of the explained covariance $\mathbf{\Sigma}$. This brings us to the second reason: we must stretch the biased projector's definition to encompass any positive definite $\mathbf{\Sigma}^{-1}$ as a *pseudo-null-steering operator*. At this point, it would be useful to think of the effect of multiplying with an inverse-covariance (precision). Assuming that we normalise the fixed effects and data such that the spectral norm $||\mathbf{\Sigma}^{-1}|| = 1$, then the spectrum of $\mathbf{\Sigma}^{-1}$ lies anywhere in $[0, 1]$, whereas the spectrum of a conventional null-steering operator (orthogonal projector) is binary (that is, it lies in $\{0, 1\}$) and only the eigenvectors of non-zero eigenvalues are intact. The added twist is that, since the magnitudes of the principal projecting directions are scaled by the *principal eigenvalues* then the pseudo-null-steering operator, and ultimately the *biased oblique projector*, can also act anywhere in between the ends of the spectrum.

Example: Dual-RCA on a toy dataset

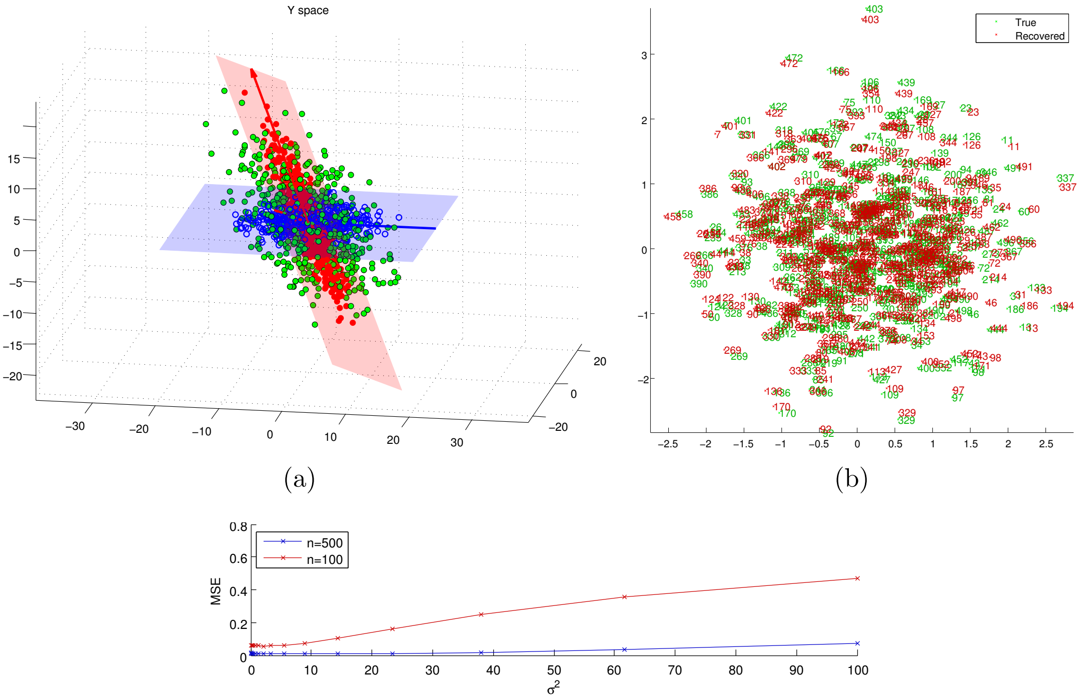
We demonstrate first a proof of concept with a 3D toy-dataset illustrated in Figure 3.4. We consider the case where variables $\mathbf{Z}$ are observed (fixed effects) or just estimated (for example, after an EM iteration) so the sample-covariance of $\mathbf{Y}$ is partially explained by the covariance of $\mathbf{Z}$ and noise. The take-away message is that RCA accounts for this covariance structure in $\mathbf{Y}$ and gives a point estimate of the latent variables $\mathbf{X}$ *up to rotation*, but with respect to the residual covariance in $\mathbf{Y}$ not explained by $\mathbf{Z}$ and noise. Another point of this example is to visualise the end result of an actual oblique projector. The reconstruction error depends on the variance σ^2 of the induced noise and the number of samples in the dataset.

3.3.3 Corollary for equivalence to PCA

In terms of objective functions, it is well known that PCA maximises the *variance* of the reduced dataset when projected on the *eigen-basis* of the sample-covariance matrix [Bishop, 2006; Hastie *et al.*, 2009; Jolliffe, 2002]; whereas *Canonical Correlation Analysis* (CCA) maximises the *correlation* of two datasets when projected on the *generalised eigenvectors* of a particular covariance structure. We review CCA in section 4.2.

RCA objective function There is an easy way to solve the GEP of RCA; it involves casting it into the equivalent form of a *regular eigenvalue problem* and then solving for the generalised eigenvectors. It also explicitly shows the objective function of RCA through a direct connection with PCA. More specifically, it follows from eq. (3.27) that

$$\begin{aligned}
\tfrac{1}{n}\,\mathbf{Y}^\top\mathbf{Y}\mathbf{S} &= \left(\mathbf{U}\boldsymbol{\Lambda}^{1/2}\right)\left(\boldsymbol{\Lambda}^{1/2}\mathbf{U}^\top\right)\mathbf{S}\mathbf{D}\\
\tfrac{1}{n}\left(\boldsymbol{\Lambda}^{-1/2}\mathbf{U}^\top\right)\mathbf{Y}^\top\mathbf{Y}\mathbf{S} &= \left(\boldsymbol{\Lambda}^{1/2}\mathbf{U}^\top\mathbf{S}\right)\mathbf{D}\\
\tfrac{1}{n}\left(\boldsymbol{\Lambda}^{-1/2}\mathbf{U}^\top\mathbf{Y}^\top\right)\left(\mathbf{Y}\mathbf{U}\boldsymbol{\Lambda}^{-1/2}\right)\widetilde{\mathbf{S}} &= \widetilde{\mathbf{S}}\mathbf{D}\\
\tfrac{1}{n}\widetilde{\mathbf{Y}}^\top\widetilde{\mathbf{Y}}\widetilde{\mathbf{S}} &= \widetilde{\mathbf{S}}\mathbf{D}\ ,
\end{aligned} \tag{3.30}$$



(a)

(b)

(c)

Figure 3.4: RCA on a 3D toy-dataset of 500 samples. Random effects **X** (blue) are hidden and fixed effects **Z** (red) are given. Each set of variables lies in a two-dimensional linear manifold in the 3D space of the observed variables **Y** (green). Each plane is spanned by its variables' principal components (arrows).
(a) Observed variables **Y** are generated through a linear combination of effects **X**, **Z** and iid Gaussian spherical noise.
(b) True values of **X** (green) and estimates by RCA (red) for $\sigma^2 = 10^{-2}$. The *Procrustes* algorithm is used on the estimated **X** for visualization purposes to find an appropriate rotation that best matches the true **X**.
(c) The mean square error of the recovered **X** as a function of the induced noise variance σ^2, for two sample sizes.

where $\widetilde{\mathbf{S}} \triangleq \mathbf{\Lambda}^{1/2}\mathbf{U}^\top\mathbf{S}$ is a transformed version of the generalised eigenvectors and $\widetilde{\mathbf{Y}}$ are the transformed data introduced in eq. (3.21). Clearly, this is the PCA eigenvalue problem on the sample-covariance of the *transformed* data. □

More generally, we have shown the corollary that:

Every RCA problem can be cast into an equivalent PCA problem.

We can append "*and vice versa*" at the end of the corollary that would work only for a fixed $\mathbf{U}$ and $\mathbf{\Lambda}$, otherwise a PCA problem can be cast into infinitely many RCA problems. But if we stick to a particular strategy[1] for inverse-transforming the data-space then we can claim that there is a bijection between the two sets of problems, thus showing that:

The sets of PCA and RCA problems are of the same size.

RCA vs simply transforming the data Tempting as it may be, we do not actually recommend solving the equivalent PCA problem from eq. (3.30), as it obviously requires an explicit transformation of the data. Note instead that RCA does not "touch" the data, a virtue which is highlighted especially when very large matrices are involved, and/or RCA is a step to some iterative scheme where preserving the sample covariance matrix is crucial to the larger algorithm, for instance, see Section 4.6 and Chapter 5.

Potential directions towards unification The above statement *guarantees* that any problem that can be cast as an RCA problem can also be solved via PCA. There is now potential to establish new connections and strengthen existing ones between classical models as well as their probabilistic counterparts, including Bayesian-linear extensions such as Bayesian-PCA [Bishop, 1999], Bayesian-CCA [Klami & Kaski, 2007; Virtanen *et al.*, 2011; Wang, 2007] and similarly through the dual-RCA for kernelised (non-linear) extensions such as the GPLVM [Lawrence, 2005], kernel-PCA [Schölkopf *et al.*, 1997] and kernel-ICA [Bach &

[1] Namely, for a fixed $\mathbf{U}$ and $\mathbf{\Lambda}$ there is a bijection between the set of covariances of size p and the set of pairs of covariances such that $\widetilde{\mathbf{Y}}^\top\widetilde{\mathbf{Y}} \mapsto \left(\mathbf{Y}^\top\mathbf{Y}, \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top\right)$ as per eq. (3.30).

Jordan, 2002]. Linear models can hardly be considered state-of-the-art and kernelised approaches are of special interest in the machine learning community as they allow us generalise classical linear methods to model non-linear relationships between datapoints with many practical applications in complex domains (e.g. biomedicine, economics).

A geometric interpretation using oblique projectors

As we mentioned in section 3.2.3, oblique projections play a important role in interpreting the proof of the RCA theorem and bring geometrical insight to the study of RCA solutions, to which we dedicate chapter 4. We review some mathematical properties of oblique projectors in appendix A.3.2.

We also saw in section 3.3.2 that the posterior expectation of RCA is strongly tied to an oblique projection as it estimates either the oblique-projected data (dual) or the projection basis (primal). The RCA theorem is novel as it introduces a probabilistic interpretation of the recovered projector subspace in the same way that PPCA enriched classical PCA for both primal and dual representations. Thereby, we distinguish the *probabilistic* interpretation of RCA from its *classical* origins in generalised projection methods, which are not as familiar in machine learning as they are to the signal processing community [Behrens & Scharf, 1994; Chen & Wang, 2006; Hu & Loizou, 2003].

In Figure 3.4 we can directly *read-off* the geometric interpretation of the operation in RCA: the eigen-basis of the fixed variables dictates the direction at which the data are projected onto the basis that explains the maximum residual variance. From a physical viewpoint, this can be seen as an oblique projection akin to a light projector displaying onto a wall at an oblique angle. The RCA proof shows us exactly the mechanics of this operation: First, the data are transformed such that the oblique angle of the "projector", and scaling thereof, are undone. Then PCA on the transformed data takes care of the projection onto the final surface.

Taking this analogy one step further — and this is where the spectral theorem really comes into play — the properties of the positive definite $\mathbf{\Sigma}^{-1}$ are uniquely characterised by its spectral decomposition. We saw this at the end of section 3.3.2, where we commented on the *pseudo-null-steering operator* $\mathbf{\Sigma}^{-1}$ not being an orthogonal projector in the strict sense, that is, not being characterised simply by its column-space (principal eigenvectors) but being enriched with more “dials” (principal eigenvalues), hence the naming. A real-world analogy would be closer to having a holographic rather than a 2-D projection. So depending on the frequencies of the “light” that we cast we can get different kinds of information from the data, as we explore in the following section.

3.4 Summary

A common approach in data modeling is to explain the behavior of an observed set of covariates through a smaller *latent* set. This motivated the study of classical models with a low-rank covariance structure plus diagonal noise (spherical or heteroscedastic). However, we are often faced with data represented by linear mixed-effects, that is, the data can be *partially explained* by a set of fixed covariates and we wish to analyse the *residual* components corresponding to the random effects in these data.

This motivated us to develop the *residual component analysis* (RCA) algorithm: a maximum-likelihood approach for describing a low dimensional representation of the residuals of a dataset given partial explanation by a fixed-effects covariance matrix $\mathbf{\Sigma}$. We showed how the low-rank component of the covariance in the marginal distribution can be determined through a generalized eigenvalue problem (GEP).

We analysed how the GEP of RCA (that is, with the joint sample-covariance as the matrix on the LHS) is essentially a regular eigenvalue problem on the joint sample-covariance (a PCA problem) of a linear transformation of the original data. Expanding on this, we showed a deeper connection based on oblique (non-orthogonal) projections of the data, where the inverse of the explained covariance term plays the role of a *null-steering operator* in the posterior expectation of the latent components.

The following chapter is dedicated on drawing connections: we will use both variants of the RCA theorem to reduce a number of probabilistic and classical low-rank models into RCA.

Chapter 4

Generalisations of classical and probabilistic models

As we show in this chapter, from the viewpoint of RCA, we can recover CCA by instantiating $\mathbf{\Sigma}$ to be block-diagonal, with each block containing the sample-covariance associated to an individual dataset, in other words, this instantiation of $\mathbf{\Sigma}$ encodes *no correlation between* the datasets but only *within*. Thereby, the generalised eigenvectors or *residual components* $\mathbf{S}$ in the GEP of RCA would explain away the remaining structure that is *not captured* by any of the sample-covariances individually — that structure being in this case the cross-correlation between the data sets. Again, we note that the framework of RCA is more generally applicable; depending on the instantiation of $\mathbf{\Sigma}$ we can explore other kinds of residual components.

4.1 Generalised eigenvalue problems

On an abstract level, problems like PCA and CCA aim to optimize some vector $\mathbf{w}$ in a metric vector space defined by $\mathbf{M}$, with some *restriction* on the solution norm in a vector space defined by $\mathbf{N}$. Problems of this general formulation lead to the maximisation of a *Rayleigh quotient* [Horn & Johnson, 1990; Parlett, 1980]:

$$\max_{\mathbf{w}} R(\mathbf{w}) \;\equiv\; \max_{\mathbf{w}} \left\{ \frac{\mathbf{w}^\top \mathbf{M}\mathbf{w}}{\mathbf{w}^\top \mathbf{N}\mathbf{w}} \right\} \; . \tag{4.1}$$

Setting the gradient of this quotient with respect to $\mathbf{w}$ to zero,

$$\nabla_{\mathbf{w}} R(\mathbf{w}) = 2\mathbf{M}\mathbf{w} - R(\mathbf{w}) 2\mathbf{N}\mathbf{w} = \mathbf{0} \; ,$$

yields the stationarity conditions expressed in the form of a GEP (not necessarily symmetric):

$$\mathbf{M}\mathbf{w} = R(\mathbf{w})\mathbf{N}\mathbf{w} \; ,$$

where the solution for $\mathbf{w}$ is obtained as the *generalised eigenvector* of $\mathbf{M}$ and $\mathbf{N}$ and the quantity of interest, initially formulated as the Rayleigh quotient, is obtained as the corresponding *generalised eigenvalue.* In settings where $\mathbf{M}$ and $\mathbf{N}$ are symmetric (e.g. covariance structures), then the generalised eigenvalues are real and the normalised generalised eigenvectors form an orthonormal basis.

4.2 A review of canonical correlation analysis

Canonical correlation analysis (CCA), originally introduced by Hotelling [1936], follows in the same track of PCA [Hotelling, 1933] in the sense that both approaches are formulated as eigenvalue problems [De Bie *et al.*, 2005].

The aim of CCA is to find weights $\mathbf{u}_1 \in \mathbb{R}^{p_1}$ and $\mathbf{u}_2 \in \mathbb{R}^{p_2}$ so as to maximize the Pearson product moment correlation between the linear combinations $\mathbf{X}\mathbf{u}_1$ and $\mathbf{Y}\mathbf{u}_2$, with the constraint that $||\mathbf{X}\mathbf{u}_1||^2 = ||\mathbf{Y}\mathbf{u}_2||^2 = 1$. A second set of solution weights can be found giving a different pair of combinations, with the added constraint that they are orthogonal to the first pair, and so on up to $\min(p_1, p_2)$ solutions. The full set of solutions is given through the GEP:

$$\begin{bmatrix} \mathbf{0} & \widehat{\mathbf{S}}_{12} \\ \widehat{\mathbf{S}}_{21} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{S}}_1 & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbf{S}}_2 \end{bmatrix} \begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{bmatrix} \mathbf{P} \; , \tag{4.2}$$

where the square block matrices contain the individual sample-covariances and cross-covariances of parts of the concatenated (joint) data

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \end{bmatrix} , \quad \text{such that} \quad \widehat{\mathbf{S}} = \begin{bmatrix} \widehat{\mathbf{S}}_1 & \widehat{\mathbf{S}}_{12} \\ \widehat{\mathbf{S}}_{21} & \widehat{\mathbf{S}}_2 \end{bmatrix} = \frac{1}{n} \begin{bmatrix} \mathbf{Y}_1^\top \mathbf{Y}_1 & \mathbf{Y}_1^\top \mathbf{Y}_2 \\ \mathbf{Y}_2^\top \mathbf{Y}_1 & \mathbf{Y}_2^\top \mathbf{Y}_2 \end{bmatrix} .$$

The generalised eigenvalues in the diagonal matrix $\mathbf{P}$ are called the *canonical correlations*. The generalised eigenvectors made up of direction-pairs $\mathbf{U}_1$ and $\mathbf{U}_2$ are known as the *canonical-directions* or *coefficients* in the data-spaces of $\mathbf{Y}_1$ and $\mathbf{Y}_2$ respectively. So these maximise the correlation between the combinations $\mathbf{Y}_1\mathbf{U}_1$ and $\mathbf{Y}_2\mathbf{U}_2$ known as *canonical variates*, such that

$$\mathbf{U}_1^\top \widehat{\mathbf{S}}_{12} \mathbf{U}_2 = \mathbf{P} \quad \text{and} \quad \mathbf{U}_1^\top \widehat{\mathbf{S}}_1 \mathbf{U}_1 = \mathbf{U}_2^\top \widehat{\mathbf{S}}_2 \mathbf{U}_2 = \mathbf{I} ,$$

where $\mathbf{P}$ now is a rectangular diagonal matrix with the canonical correlations on its diagonal.

PCCA Bach & Jordan [2002] showed that the *probabilistic-CCA* model, for centered data

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{V}_1\mathbf{V}_1^\top & \mathbf{V}_1\mathbf{V}_2^\top \\ \mathbf{V}_2\mathbf{V}_1^\top & \mathbf{V}_2\mathbf{V}_2^\top \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_2 \end{bmatrix} \right) , \tag{4.3}$$

illustrated in Figure 3.3(b), p. 45, has the maximum-likelihood solution[1]:

$$\begin{bmatrix} \widehat{\mathbf{V}}_1 \\ \widehat{\mathbf{V}}_2 \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{S}}_1 & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbf{S}}_2 \end{bmatrix} \begin{bmatrix} \mathbf{U}_{1q} \\ \mathbf{U}_{2q} \end{bmatrix} \mathbf{P}_q^{1/2} \mathbf{R} \tag{4.4}$$

$$\begin{aligned} \widehat{\boldsymbol{\Sigma}}_1 &= \widehat{\mathbf{S}}_1 - \widehat{\mathbf{V}}_1 \widehat{\mathbf{V}}_1^\top \\ \widehat{\boldsymbol{\Sigma}}_2 &= \widehat{\mathbf{S}}_2 - \widehat{\mathbf{V}}_2 \widehat{\mathbf{V}}_2^\top , \end{aligned} \tag{4.5}$$

where $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are full noise covariance matrices, $\mathbf{U}_{1q}$ and $\mathbf{U}_{2q}$ are the first q pairs of *canonical directions*, $\mathbf{P}_q$ is the diagonal matrix of the first q *canonical correlations* and $\mathbf{R}$ is an arbitrary rotation matrix that can be set as the identity

[1] The solution that maximises the conditional entropy of $\mathbf{y}|\mathbf{z}$.

for simplicity.

4.3 CCA, PCA and factor analysis as RCA

CCA Loosely speaking, since the task of CCA is to capture only the *linearly shared* structure between two sets of variables and treat all other structure as noise, then in principle we can reproduce this in RCA by canceling any structure captured *within* the datasets and focus on the *shared* or *residual* structure, that is, on the linear mechanisms that cause the two sets of covariates to respond *similarly.* This RCA-like interpretation of CCA, allows us to perform tasks involving shared and private structures otherwise impossible with classical CCA; for instance, in section 5.2 we show how to explain away with dual-RCA some estimated *shared* structure between paired times-series of two experimental conditions and focus on the *differential* structure; in section 4.5 we “re-invent” the learning of a multi-view model in statistics known as *inter-battery factor analysis* [IBFA, Browne, 1979; Tucker, 1958] and *extended-CCA* in machine learning [Klami & Kaski, 2006, 2008].

To show how RCA can be reduced to CCA, we compare their GEPs: recall that the GEP of primal RCA is

$$\tfrac{1}{n}\mathbf{Y}^\top\mathbf{Y}\mathbf{S} = \mathbf{\Sigma}\mathbf{S}\mathbf{D}$$

and with very little algebra eq. (4.2) can be re-expressed as

$$\begin{bmatrix} \widehat{\mathbf{S}}_{11} & \widehat{\mathbf{S}}_{12} \\ \widehat{\mathbf{S}}_{12}^\top & \widehat{\mathbf{S}}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{S}}_{11} & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbf{S}}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{bmatrix} (\mathbf{P}+\mathbf{I}) \ . \tag{4.6}$$

By inspection, we can clearly see the RCA-view of CCA: the canonical directions $\mathbf{U}$ of CCA are recovered as the *generalised eigenvectors* $\mathbf{S}$ of RCA and the corresponding canonical correlations as the *shifted generalised eigenvalues* $\mathbf{P} = \mathbf{D} - \mathbf{I}$.

In other words, the instantiation

$$\mathbf{\Sigma}_{CCA} = \tfrac{1}{n}\begin{bmatrix}\mathbf{Y}_1^\top\mathbf{Y}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{Y}_2^\top\mathbf{Y}_2\end{bmatrix} = \begin{bmatrix}\widehat{\mathbf{S}}_{11} & \mathbf{0}\\ \mathbf{0} & \widehat{\mathbf{S}}_{22}\end{bmatrix}\ , \tag{4.7}$$

reduces the GEP of RCA to that of CCA. □

Therefore, the RCA solution from eq. (3.26) (reproduced here for convenience)

$$\widehat{\mathbf{W}} = \mathbf{\Sigma}\mathbf{S}(\mathbf{D}-\mathbf{I})^{1/2}\ ,$$

reduces to the PCCA maximum-likelihood solution from eq. (4.4) for

$$\mathbf{W} = \begin{bmatrix}\mathbf{V}_1\\ \mathbf{V}_2\end{bmatrix}\ ,\quad \mathbf{\Sigma} = \begin{bmatrix}\widehat{\mathbf{S}}_{11} & \mathbf{0}\\ \mathbf{0} & \widehat{\mathbf{S}}_{22}\end{bmatrix}\ ,\quad \mathbf{S} = \begin{bmatrix}\mathbf{U}_1\\ \mathbf{U}_2\end{bmatrix}\ ,\quad \mathbf{D} = \mathbf{P}+\mathbf{I}\ .$$

Note that the canonical correlations in $\mathbf{P}$ always lie in the range $[-1, 1]$, so the eigenvalues in $\mathbf{D}$ always lie in $[0, 2]$.

PCA Similarly, we get the PCA eigenvalue problem for $\mathbf{\Sigma} = \mathbf{I}$ and the PPCA maximum-likelihood solution from eq. (3.3) is recovered as

$$\widehat{\mathbf{W}} = \mathbf{S}_q(\mathbf{D}_q-\mathbf{I})^{1/2} = \mathbf{U}_q(\mathbf{\Lambda}_q-\mathbf{I})^{1/2}\ . \qquad \square$$

From this analysis we can conclude that from the RCA viewpoint, CCA can be seen as setting $\mathbf{\Sigma}$ as a block-diagonal covariance matrix, with each block containing the sample-covariance associated to an individual dataset, or more intuitively, a $\mathbf{\Sigma}$ instantiation that encodes *a lack of correlation* between the two datasets. Consequently, the residual components in $\mathbf{S}$ are forced to capture the correlation structure *between* the data sets, which is the *residual* structure missed by the individual sample-covariances. In the case of PCA, no structure in the covariance is explained ($\mathbf{\Sigma}$ is the identity), therefore all of the structure remains to be captured by the *principal eigenvectors* of the sample-covariance $\widehat{\mathbf{S}}$. The analogue equivalences for the dual representations are directly parallel to the primal representations discussed here.

Factor analysis Recall that the covariance structure of a *factor analysis* model [Bartholomew *et al.*, 2011; Basilevsky, 1994] is slightly more general than that of PPCA; the marginal covariance has the form of a *low-rank plus diagonal noise* structure:

$$\mathbf{y} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{W}\mathbf{W}^\top + \operatorname{diag}(\boldsymbol{\alpha})\right) ,$$

where $\operatorname{diag}(\boldsymbol{\alpha})$ is a diagonal matrix comprised of the noise variances in $\boldsymbol{\alpha}$. The *heteroscedasticity* of the noise, on the one hand, makes the factor analysis model more flexible than PPCA while maintaining the convexity of the likelihood function, but on the other hand introduces interaction terms in the stationarity (zero gradient) conditions between the components of $\mathbf{W}$ and $\boldsymbol{\alpha}$, so the maximum likelihood estimates of the parameters $(\mathbf{W}, \boldsymbol{\alpha})$ must be obtained iteratively, the simplest way being via the expectation-maximisation (EM) algorithm [Rubin & Thayer, 1982]. In practice, the noise variances $\boldsymbol{\alpha}$ are not known a priori, though they can be fixed as part of the E-steps during an EM run and the solution for $\mathbf{W}$ conditioned on $\boldsymbol{\alpha}$ has exactly the form of the RCA solution:

$$\widehat{\mathbf{W}} = \boldsymbol{\Sigma}_{FA}(\mathbf{D}_q - \mathbf{I})^{1/2} ,$$

when the explained covariance term is $\boldsymbol{\Sigma}_{FA} = \operatorname{diag}(\boldsymbol{\alpha})$.

Casting CCA as an RCA problem is a interesting result because other generalisations and connection drawn from the CCA literature readily follow into our framework as we show in the following sections. Nonetheless, we emphasise that these are special cases and the framework of residual component analysis is more general. Later in the chapter, we see that by alternative choices for $\boldsymbol{\Sigma}$ we can explore other kinds of residual components with practical applications.

4.4 LDA as RCA

Linear discriminant analysis (LDA) or *multiple discriminant analysis* is a linear dimensionality reduction approach to multi-class classification [Duda & Hart, 1973; Fukunaga, 1990]. It is also known as *Fisher discriminant analysis* when restricted to binary classification settings [Fisher, 1936]. Assuming the n datapoints are centered and split across a total of k classes $\{\mathcal{C}_c\}_{1..k}$ with $n_c \triangleq |\mathcal{C}_c|$ memberships and class mean $\mathbf{m}_c$, then the *between class* covariance is defined as

$$\widehat{\mathbf{S}}_B \triangleq \frac{1}{n} \sum_c n_c \; \mathbf{m}_c \mathbf{m}_c^\top = \sum_c \pi_c \; \mathbf{m}_c \mathbf{m}_c^\top = \mathbf{M} \operatorname{diag}(\boldsymbol{\pi}) \mathbf{M}^\top \ ,$$

where $\mathbf{M} = [\mathbf{m}_1 \ldots \mathbf{m}_k]$, and the *within class* covariance is defined as the weighted average over all covariance-per-class matrices $\widehat{\mathbf{S}}_c = \frac{1}{n_c} \sum_{i \in \mathcal{C}_c} (\mathbf{x}_i - \mathbf{m}_c)(\mathbf{x}_i - \mathbf{m}_c)^\top$:

$$\widehat{\mathbf{S}}_W \quad \triangleq \quad \sum_c \pi_c \widehat{\mathbf{S}}_c \quad = \quad \widehat{\mathbf{S}} - \widehat{\mathbf{S}}_B \ .$$

The aim of LDA is to project the data on a hyperplane $\mathbf{u}$ residing in the data space, so as to maximise the *between-class* projected covariance $\mathbf{u}^\top \widehat{\mathbf{S}}_B \mathbf{u}$ as a measure of class separation, such that the *within-class* projected covariance $\mathbf{u}^\top \widehat{\mathbf{S}}_W \mathbf{u}$ is also minimised to prevent further class overlap on the projecting hyperplane. From our discussion on GEPs in beginning of this section, the above description can be naturally formalised as the maximisation of a Rayleigh quotient:

$$\widehat{\mathbf{u}} = \max_{\mathbf{u}} \left\{ \frac{\mathbf{w}^\top \widehat{\mathbf{S}}_B \mathbf{u}}{\mathbf{u}^\top \widehat{\mathbf{S}}_W \mathbf{u}} \right\} \ ,$$

whose complete set of solutions is given by the GEP:

$$\widehat{\mathbf{S}}_B \mathbf{U} = \widehat{\mathbf{S}}_W \mathbf{U} \mathbf{P} \ . \tag{4.8}$$

As an interesting marriage of *supervised* and *unsupervised learning*, it is also well known that the same LDA solution is obtained via CCA [Bach & Jordan, 2002; De Bie *et al.*, 2005; Sun *et al.*, 2011] when the centered dataset $\mathbf{Y}$ is coupled with the *target* matrix $\mathbf{T}$ whose rows are the target vectors in the 1-of-k encoding[1]:

$$\begin{bmatrix} \mathbf{0} & \widehat{\mathbf{S}}_{\mathbf{YT}} \\ \widehat{\mathbf{S}}_{\mathbf{TY}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{\mathbf{Y}} \\ \mathbf{U}_{\mathbf{T}} \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{S}}_{\mathbf{YY}} & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbf{S}}_{\mathbf{TT}} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{\mathbf{Y}} \\ \mathbf{U}_{\mathbf{T}} \end{bmatrix} \mathbf{P}' \ , \tag{4.9}$$

[1] A datapoint $\mathbf{y}_i$ belonging in class $\mathcal{C}_3$ out of 5 classes is paired to the target vector $\mathbf{t}_i = (0, 0, 1, 0, 0)^\top$.

where the blocks are the covariances and cross-covariances of the two datasets, $\mathbf{U}_{\mathbf{Y}}$ equals $\mathbf{U}$ from eq. (4.8) and $\widehat{\mathbf{S}}_{\mathbf{YY}} = \widehat{\mathbf{S}}$.

One would justifiably think that since the set of RCA problems contains the set of CCA problems, which in turn contains the set of LDA problems, then RCA subsumes LDA. Indeed, as we show RCA can be reduced directly to either solution form. In eq. (4.8) different algebraic manipulations give GEPs with different eigenvalues; more specifically, adding:

$$\begin{aligned} \widehat{\mathbf{S}}_W\mathbf{U} \text{ to both sides gives} \quad & \widehat{\mathbf{S}}\mathbf{U} = \widehat{\mathbf{S}}_W\mathbf{U}(\mathbf{P}+\mathbf{I}) \ , & (4.10) \\ \text{or } \widehat{\mathbf{S}}_B\mathbf{U}\mathbf{P} \text{ to both sides gives} \quad & \widehat{\mathbf{S}}\mathbf{U} = \widehat{\mathbf{S}}_B\mathbf{U}(\mathbf{P}^{-1}+\mathbf{I}) \ , & (4.11) \end{aligned}$$

(recall that $\widehat{\mathbf{S}} = \widehat{\mathbf{S}}_B + \widehat{\mathbf{S}}_W$). Especially from eq. (4.10), now we can read the RCA interpretation of LDA, in which the explained structure is the within-class covariance and the generalised eigenvectors of RCA (typically the first $k-1$) are the required discriminants onto which the data are projected. From the CCA view in eq. (4.9), we repeat the same algebraic manipulation that we used for reducing any CCA problem to RCA:

$$\begin{bmatrix} \widehat{\mathbf{S}}_{\mathbf{YY}} & \widehat{\mathbf{S}}_{\mathbf{YT}} \\ \widehat{\mathbf{S}}_{\mathbf{TY}} & \widehat{\mathbf{S}}_{\mathbf{TT}} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{\mathbf{Y}} \\ \mathbf{U}_{\mathbf{T}} \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{S}}_{\mathbf{YY}} & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbf{S}}_{\mathbf{TT}} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{\mathbf{Y}} \\ \mathbf{U}_{\mathbf{T}} \end{bmatrix} (\mathbf{P}'+\mathbf{I}) \ ,$$

Potential for spectral clustering It is worth noting that the CCA view in eq. (4.9) makes explicit use of label data in $\mathbf{T}$ and the discriminant directions capture the maximum correlation between the paired datasets, whereas the RCA view makes explicit use of *non-class* information encoded in the within-class covariance $\widehat{\mathbf{S}}_W$ as a proxy to uncovering the class-related structure in the sample-covariance. Both forms give the same solution and show intuitive *spectral classification* algorithms. However, eq. (4.11) and its dual counterpart can be potentially extended for *spectral clustering* [Azar *et al.*, 2001; Kannan *et al.*, 2004; Ng *et al.*, 2002] where the between-class covariance $\widehat{\mathbf{S}}_B$ is unknown and optimised in a constrained fashion (e.g. as a sparse positive definite).

4.5 Revisiting some generalisations of CCA

Bach & Jordan [2002] proposed a *kernelised* version of CCA for computing a *contrast function* of a set of non-Gaussian random variables in the form of a non-linear correlation measure in a *reproducing kernel Hilbert space* (RKHS). This contrast function is minimised as a proxy to the *mutual information* of the observed variables which amounts to *independent component analysis* (ICA) on non-Gaussian-distributed data. In the same paper the authors also explored relationships between CCA and mutual information and the generalisations of CCA to more than two sets of variables. Other instantiations of kernel-CCA restricted to two multivariate random variables were proposed by Lai & Fyfe [2001] and Akaho [2001]. Since CCA is a special case of RCA, it is natural to ask about fitting these developments into our framework and the potential directions they might lead to.

4.5.1 RCA and mutual information

It is well known that the mutual information between two multivariate Gaussian random variables $y_1 \in \mathbb{R}^{p_1}$ and $y_2 \in \mathbb{R}^{p_2}$ with joint covariance $\mathbf{S} = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{21} & \mathbf{S}_{22} \end{bmatrix}$ can be computed exactly:

$$M(\mathbf{y}_1, \mathbf{y}_2) = -\frac{1}{2} \log \frac{|\mathbf{S}|}{|\mathbf{S}_{11}|\ |\mathbf{S}_{22}|} \ ,$$

and that the fraction of determinants is equal to the product of the *canonical correlations* of CCA on $\mathbf{y}_1$ and $\mathbf{y}_2$. This is because the fraction of determinants equals the determinant of the covariance-product in the eigenvalue problem (LHS):

$$\begin{bmatrix} \mathbf{S}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{22} \end{bmatrix}^{-1} \mathbf{S}\mathbf{U} = \mathbf{U}(\mathbf{P} + \mathbf{I}) \ ,$$

which shares the same eigenvalues with the equivalent GEP of CCA

$$\mathbf{S}\mathbf{U} = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{22} \end{bmatrix} \mathbf{U}(\mathbf{P}+\mathbf{I}) \; .$$

Therefore $M(\mathbf{y}_1, \mathbf{y}_2) = -\frac{1}{2}\log|\mathbf{P}+\mathbf{I}|$, assuming $\mathbf{P}$ is square with size $\min(p_1, p_2)$.

4.5.2 Towards massive-view learning through RCA

In fact when Bach & Jordan [2002] showed the above, they used the RCA view of CCA (to express it in term of the joint covariance matrix). However, the authors did not comment on the generality of this form (that is, beyond the significance of the block-diagonal on the RHS as the *explained covariance*). Kettenring [1971] was the first to consider various extensions of CCA to more than two datasets. However, the one proposed by Bach & Jordan [2002, appendix A.2], simply by expanding the block-diagonal for more datasets,

$$\begin{bmatrix} \mathbf{S}_{11} & \dots & \mathbf{S}_{1m} \\ \vdots & \ddots & \vdots \\ \mathbf{S}_{m1} & \dots & \mathbf{S}_{mm} \end{bmatrix} \begin{bmatrix} \mathbf{U}_1 \\ \vdots \\ \mathbf{U}_m \end{bmatrix} = \begin{bmatrix} \mathbf{S}_{11} & & \\ & \ddots & \\ & & \mathbf{S}_{mm} \end{bmatrix} \begin{bmatrix} \mathbf{U}_1 \\ \vdots \\ \mathbf{U}_m \end{bmatrix} \mathbf{P} \; ,$$

fits naturally into the RCA framework, as the explained covariance term cancels out any structure that jointly treats the datasets as pairwise uncorrelated. As expected, the mutual information between many multivariate Gaussian random variables generalises just as easily

$$M(\mathbf{y}_1, \dots, \mathbf{y}_m) = -\frac{1}{2}\log\frac{|\mathbf{S}|}{|\mathbf{S}_{11}|\dots|\mathbf{S}_{mm}|} \; .$$

This connection strengthens the justification for using the canonical correlations in a RKHS as a proxy for minimising the mutual information between sets of non-Gaussian random variables, which in general is a function of higher-order moments of their true distributions and not just correlation (second-order) in the primal space. Furthermore, in many applications probabilistic-CCA is used as a basis for *multi-view learning* of a single latent multivariate random variable (usually with the addition of view-specific latent variables). The above generalisation to *many* variables through the *RCA interpretation* could provide a basis for *massive-view learning*, that is, scenarios where many different datasets with aligned[1] samples (e.g. a gene expression microarray experiment performed in many different labs around the world, or measurements from multiple sensors scattered in region) can potentially provide deeper insight on the fundamental factors of common variation.

Note that by diagonalising the sample-covariance of each dataset a priori (reducing the block-diagonal to a diagonal $\mathbf{\Sigma}$, hence reducing the algorithm to PCA or FA) would be detrimental to the massive-view aspect of the algorithm. So *massive-view RCA* can be seen as a *factor analysis* approach on the level of data-*sets* as opposed to the conventional level of data-*points*, the only requirement being that the datasets have the same number of samples (for the primal representation) or the same number of features (dual).

4.6 An algorithm for inter-battery factor analysis

Probabilistic canonical correlation analysis (PCCA, sec. 4.2) models the covariance structure of two paired datasets with a full-rank block-diagonal and low-rank off-diagonal terms, see eq. (4.3), p. 66. Tucker [1958] introduced *inter-battery factor analysis* (IBFA) which extends classical CCA with *view-specific* components (in addition to the standard components shared by both views). IBFA is a more realistic model for multi-view learning as it attempts to explain data also with components exclusive to each dataset. Figure 3.3(c), p. 45, shows the graphical model of IBFA. In the statistics literature, Browne [1979] worked out a maximum-likelihood algorithm for learning IBFA. In the machine learning literature, Klami & Kaski [2006, 2008] independently developed an EM approach to learning IBFA and a Bayesian approach to automatically learn the latent dimensionalities [Klami

[1] Meaning that the correspondence of any particular sample across the datasets is known.

& Kaski, 2007] and group components into shared and view-specific sets [Virtanen *et al.*, 2011] via ARD priors. Ek *et al.* [2008] proposed a non-linear version of IBFA which maps the shared and private (view-specific) latent spaces to the observed space through *Gaussian processes.*

Each dataset, $\mathbf{Y}_d \in \mathbb{R}^{n \times p_d}$, $d \in \{1, 2\}$, associates to its own set of latent points, $\mathbf{X}_d \in \mathbb{R}^{n \times q_d}$ as well as the shared latent points, $\mathbf{Z} \in \mathbb{R}^{n \times q}$, that lie in the shared latent space found in classical CCA. As an advantage of this structure, if the covariance specific to each dataset is low-rank then this will be recovered. The data partition is represented as

$$\begin{aligned} \mathbf{y}_1 &= \mathbf{W}_1\mathbf{x}_1 + \mathbf{V}_1\mathbf{z} + \boldsymbol{\epsilon}_1 \quad \text{with noise} \quad \boldsymbol{\epsilon}_1 \sim \mathcal{N}(\mathbf{0}, \sigma_1\mathbf{I}) \quad \text{and} \\ \mathbf{y}_2 &= \mathbf{W}_2\mathbf{x}_2 + \mathbf{V}_2\mathbf{z} + \boldsymbol{\epsilon}_2 \quad \text{with noise} \quad \boldsymbol{\epsilon}_2 \sim \mathcal{N}(\mathbf{0}, \sigma_2\mathbf{I}) \ . \end{aligned}$$

Each set of latent variables is marginalized through an isotropic Gaussian prior to give a marginal covariance structure for the concatenated data

$$\mathbf{S} = \begin{bmatrix} \mathbf{W}_1\mathbf{W}_1^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_2\mathbf{W}_2^\top \end{bmatrix} + \begin{bmatrix} \mathbf{V}_1\mathbf{V}_1^\top & \mathbf{V}_1\mathbf{V}_2^\top \\ \mathbf{V}_2\mathbf{V}_1^\top & \mathbf{V}_2\mathbf{V}_2^\top \end{bmatrix} + \begin{bmatrix} \sigma_1^2\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \sigma_2^2\mathbf{I} \end{bmatrix} .$$

On the one hand, if the view-specific weights $\mathbf{W}_1$ and $\mathbf{W}_2$ are known, we can learn the shared shared-view weights, $\mathbf{V}^\top = [\mathbf{V}_1^\top \ \mathbf{V}_2^\top]$, thought the RCA algorithm by setting the explained covariance term as

$$\boldsymbol{\Sigma}_{IBFA} = \begin{bmatrix} \mathbf{W}_1\mathbf{W}_1^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_2\mathbf{W}_2^\top \end{bmatrix} + \begin{bmatrix} \sigma_1^2\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \sigma_2^2\mathbf{I} \end{bmatrix} .$$

On the other hand, to learn $\mathbf{W}_1$ and $\mathbf{W}_2$ we note that the marginal covariance of $\mathbf{y}_1$ is the block $\mathbf{S}_{11} = \mathbf{W}_1\mathbf{W}_1^\top + \mathbf{V}_1\mathbf{V}_1^\top + \sigma_1^2\mathbf{I}$. So if $\mathbf{V}_1$ is known, we can learn $\mathbf{W}_1$ with RCA using $\boldsymbol{\Sigma} = \mathbf{V}_1\mathbf{V}_1^\top + \sigma_1^2\mathbf{I}$. We follow an analogous procedure for learning $\mathbf{W}_2$.

One obvious question with IBFA is how to choose the latent dimensionalities. If the noise variance σ^2 is fixed in probabilistic PCA then the latent dimension q is determined automatically by choosing the maximal set of q principal components $\mathbf{W}^{(q)}$ such that $\lambda_q > \sigma^2$, see eq. (3.4), p. 39. This reduces the problem of choosing the *intrinsic dimension* to choosing a suitable *noise level*. We follow a similar approach with iterative-RCA by setting the noise variances to a fraction $\alpha \in [0, 1]$ of the total data variance $\mathrm{tr}\,(\mathbf{S})$ and tune α to control the latent dimensionality. The algorithm converges when the log-marginal likelihood drops less than a small constant (10^{-6}). Algorithm 1 lists one variant of iterative-RCA:

Algorithm 1 iterative-RCA

```
Initialize $\alpha \in [0,1]$;   $\widehat{\mathbf{S}} = \begin{bmatrix} \widehat{\mathbf{S}}_{11} & \widehat{\mathbf{S}}_{12} \\ \widehat{\mathbf{S}}_{21} & \widehat{\mathbf{S}}_{22} \end{bmatrix} \leftarrow \frac{1}{n} \begin{bmatrix} \mathbf{Y}_1^\top \mathbf{Y}_1 & \mathbf{Y}_1^\top \mathbf{Y}_2 \\ \mathbf{Y}_2^\top \mathbf{Y}_1 & \mathbf{Y}_2^\top \mathbf{Y}_2 \end{bmatrix}$
$\sigma_1^2 \leftarrow \frac{\alpha}{p_1} \mathrm{tr} \left( \widehat{\mathbf{S}}_{11} \right)$;   $\sigma_2^2 \leftarrow \frac{\alpha}{p_2} \mathrm{tr} \left( \widehat{\mathbf{S}}_{22} \right)$;   $\widehat{\mathbf{W}}_1 \leftarrow \widehat{\mathbf{V}}_1 \leftarrow \mathbf{0}_{p_1}$;   $\widehat{\mathbf{W}}_2 \leftarrow \widehat{\mathbf{V}}_2 \leftarrow \mathbf{0}_{p_2}$

repeat
  View-specific step:
    Solve for  $\widetilde{\mathbf{W}}_1$  in  $\widehat{\mathbf{S}}_{11} \widetilde{\mathbf{W}}_1 = (\widehat{\mathbf{V}}_1 \widehat{\mathbf{V}}_1^\top + \sigma_1^2 \mathbf{I})\, \widetilde{\mathbf{W}}_1 \mathbf{\Lambda}_1$
    $\widehat{\mathbf{W}}_1 \leftarrow (\widehat{\mathbf{V}}_1 \widehat{\mathbf{V}}_1^\top + \sigma_1^2 \mathbf{I})\, \widetilde{\mathbf{W}}_1^{(q_1)} (\mathbf{\Lambda}_1^{(q_1)} - \mathbf{I})^{1/2}$

    Solve for  $\widetilde{\mathbf{W}}_2$  in  $\widehat{\mathbf{S}}_{22} \widetilde{\mathbf{W}}_2 = (\widehat{\mathbf{V}}_2 \widehat{\mathbf{V}}_2^\top + \sigma_2^2 \mathbf{I}) \widetilde{\mathbf{W}}_2 \mathbf{\Lambda}_2$
    $\widehat{\mathbf{W}}_2 \leftarrow (\widehat{\mathbf{V}}_2 \widehat{\mathbf{V}}_2^\top + \sigma_2^2 \mathbf{I}) \widetilde{\mathbf{W}}_2^{(q)} (\mathbf{\Lambda}_2^{(q)} - \mathbf{I})^{1/2}$

  View-shared step:
    $\mathbf{\Sigma} \leftarrow \begin{bmatrix} \widehat{\mathbf{W}}_1 \widehat{\mathbf{W}}_1^\top + \sigma_1^2 \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbf{W}}_2 \widehat{\mathbf{W}}_2^\top + \sigma_2^2 \mathbf{I} \end{bmatrix}$
    Solve for  $\widetilde{\mathbf{V}}$  in  $\widehat{\mathbf{S}} \widetilde{\mathbf{V}} = \mathbf{\Sigma} \widetilde{\mathbf{V}} \mathbf{\Lambda}$
    $\widehat{\mathbf{V}} \leftarrow \mathbf{\Sigma} \widetilde{\mathbf{V}}^{(q)} (\mathbf{\Lambda}^{(q)} - \mathbf{I})^{1/2}$

until the log-marginal likelihood converges.
```

4.6.1 An example on pose recovery

We experiment on motion capture data produced by Agarwal & Triggs [2006] to demonstrate the effect of learning view-specific components through *iterative-RCA*. The data contain $n = 1,927$ frames of human poses (3D point clouds), each paired to a 2D silhouette. Each pose is represented by 21 sensors with 3D coordinates, giving $p_1 = 63$ features, and the pose data are collected in $\mathbf{Y}_1 \in \mathbb{R}^{n \times p_1}$.

Each silhouette is summarized by $p_2 = 100$ HoG[1] features and the silhouette data are collected in $\mathbf{Y}_2 \in \mathbb{R}^{n \times p_2}$. Because both datasets were produced in the studio, we add a small amount of iid Gaussian spherical noise to each feature to simulate conditions closer to the outside world.

Our task is to predict the pose $\mathbf{y}_1$ of a silhouette $\mathbf{y}_2$. The posterior (predictive) mean of $p(\mathbf{y}_1^*|\mathbf{y}_2)$ is:

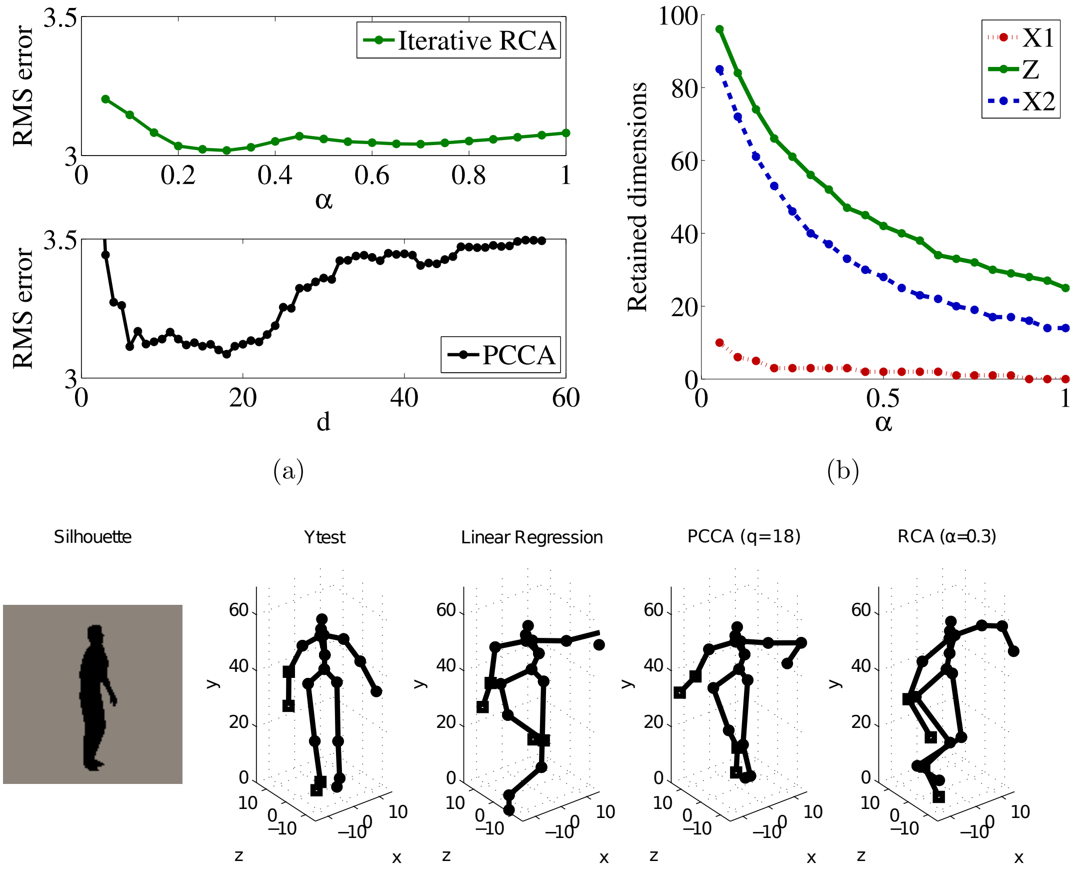
$$\mathbb{E}\left[\mathbf{y}_1^*|\mathbf{y}_2\right] = \mathbf{V}_1\mathbf{V}_2^\top(\mathbf{W}_2\mathbf{W}_2^\top + \sigma_2^2)^{-1}\mathbf{y}_2 + \boldsymbol{\mu}_1 \ ,$$

where $\boldsymbol{\mu}_1$ is the sample mean of $\mathbf{Y}_2$. Posterior variances are not required for this experiment. Figure 4.1(a) compares the prediction root mean square errors (RMSE) of iterative-RCA and probabilistic CCA while varying α or q (one determines the other). Iterative-RCA generally outperforms PCCA, with the smallest difference being at $q = 18$ for PCCA (or $\alpha = 0.3$ for RCA). The RMSE of RCA is robust for a wide range of large α values. An interesting aspect of iterative-RCA is the self-regularity that the noise variance imposes on the latent dimensionality of the shared and view-specific components: For example, Figure 4.1(b) shows the increase of noise with α causing the eigenvalues to decay faster from $\mathbf{z}$ and $\mathbf{x}_2$ than from $\mathbf{x}_1$. Trimming the latent dimensionality by explaining part of the variance as noise was simple enough for illustration, but more principled approaches to dimension selection can be followed (for instance, through the BIC criterion [Schwarz, 1978]).

4.7 Summary

We discussed how the RCA theorem (both primal and dual representations) provides a probabilistic interpretation of generalised-projection low-rank models and as such can potentially unify many different algorithms. For instance, we saw *probabilistic-PCA* and *probabilistic-CCA* arise as special cases of our algorithm. The same cannot be said about LDA as the class labels or cluster assignments are non-Gaussian in general, unless a continuous relaxation of the labels is assumed.

[1] Dalal & Triggs [2005].



(a)

(b)

(c)

Figure 4.1: Comparison of iterative-RCA with probabilistic CCA shows the merit of accounting for view-specific components. (a) RMSE (across all test frames) of iterative-RCA and PCCA on reconstructing poses from silhouettes. The figure shows the error as a function of latent dimension q for PCCA and α (the fraction of explained variance) for RCA. Linear regression (not visible) yields RMSE = 3.21. (b) Latent dimensions of $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{z}$ after convergence as functions of α. (c) Test frame #404 reports the largest errors in the test set. Figure shows the test silhouette and paired true pose followed by the predicted poses.

Alternatively, those parts of the data can be “Gaussianised” through Gaussian copulas. Nonetheless, if we dispense with any probabilistic notion in RCA, its GEP still reduces to that of LDA for a special choice of $\mathbf{\Sigma}$. At numerous points across this chapter, we have shown that with further imaginative instantiations of $\mathbf{\Sigma}$ we can develop new approaches to data analysis.

In the following chapter we begin to flesh out the applications of RCA hinted in section 3.2.1.

The main idea will be a sum of low-rank and sparse inverse-covariance structures, with promising application in computational biology for heterogeneous data with hidden confounders. To link with chapter 2, a secondary application example will be given on analysing residuals left from a GP.

Chapter 5

Applications of RCA

Whereas the previous chapter focused on drawing links; here we aim to show the applicability of the RCA framework. The centerpiece of this chapter is the composite structure of *low-rank plus sparse-inverse* covariance. To motivate this structure, section 5.1 starts off with a formal description of the problem — a linear mixed-effects equation, where one effect is hidden and low-dimensional and the other is Gaussian-distributed with a sparse precision — and slowly introduces the pieces of the composite structure that will help us identify this equation. Section 5.1.1 describes our proposed methodology and section 5.1.2 describes a number of experiments across a number of application domains. A second application of RCA on the analysis of residuals of a GP is given in section 5.2, that also provides a link to the first chapter of the thesis.

5.1 Accounting for confounders in sparse Gaussian Markov random fields

Consider the following *linear mixed-effects* model [Pinheiro & Bates, 2000] of observed centered covariates $\mathbf{y} \in \mathbb{R}^p$ with two sets of factors $\mathbf{x} \in \mathbb{R}^q$, $\mathbf{z} \in \mathbb{R}^p$ and noise $\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \mathbf{I}_p\right)$:

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{z} + \boldsymbol{\epsilon} \ , \tag{5.1}$$

where $\mathbf{z}$ is a vector of "*fixed*" effects either in the form of known, given or estimated covariates postulated to have some predictive power over $\mathbf{y}$ (in practice they are *unknown*, but eventually they are *fixed* to some estimate to determine the residual components) and $\mathbf{x}$ is a lower dimensional vector of random effects (in general we assume there are $q < p$ latent variables). The graphical model of this representation is illustrated in Figure 3.3(a), p. 45. The focus of this chapter is a particular case of eq. (5.1), in which the "*fixed*" covariates $\mathbf{z}$ are distributed according to a zero-mean Gaussian with a *sparse-inverse* covariance (or sparse precision) matrix. This is a potentially powerful representation for many application domains, as we demonstrate with examples on computational biology, motion-capture and social networks.

Consider a case where samples $\mathbf{y}_i \in \mathbb{R}^p$ are heterogeneous (for instance, a set of activation signals from p proteins, measured under various external stimuli). In a *primal* representation we aim to recover a sparse network of interactions between the *features* (proteins). The representation of $\mathbf{y}$ depends on two types of background effects: in $\mathbf{z}$ there are as many factors as there are observed covariates in $\mathbf{y}$. The factors in $\mathbf{z}$ are special in the sense that they share a *sparse* network of *conditional dependencies*. Sparse dependencies are interesting for learning parsimonious models (in this case, a protein regulation network) but in realistic scenarios this sparsity is *confounded* by the heterogeneous experimental conditions (the various stimuli) under which each sample of $\mathbf{y}$ is generated. In a sense, $\mathbf{z}$ represents a "clean", *unconfounded*, version of $\mathbf{y}$. We encode these *confounders* with the second type of background effects $\mathbf{x}$. The hypothesis is that, if there are any underlying confounding effects in the data generation process, these are fewer than our observed covariates and the confounders somehow combine in a linear fashion, $\mathbf{Wx}$, to affect an observed covariate (protein signal). Then, the structure explained solely by $\mathbf{x}$ corresponds to a *low-rank* term in the marginal covariance, see the primal version of eq. (3.13), p. 46. Another intuitive way to argue for modeling the confounding structure with a low-rank component, is by seeing how the nominal values of our signals are *forced* to diverge from their otherwise true values: because there are always fewer confounders than observed covariates, there is *redundancy* in their expression on the observed space, thus adding a low-rank structure in the covariance of our measurements.

Since by assumption the regulatory network is sparse, we parameterise the explained covariance term as:

$$\mathbf{\Sigma}_{GMRF}{}^{1} = \mathbf{\Lambda}^{-1} \ , \tag{5.2}$$

where $\mathbf{\Lambda}$ is a *sparse* positive definite matrix, thus recovering a *low-rank plus sparse-inverse* parameterisation of the covariance in eq. (3.13), p. 46. It is well known, that the precision matrix $\mathbf{\Lambda}$ of a multivariate Gaussian distribution has elements $\Lambda_{ij} = 0$ *if and only if* the variables i and j are independent when conditioned on the rest [Lauritzen, 1996]. Thereby, a sparse precision induces a sparsely connected *Gaussian Markov random field* (GMRF) or *Gaussian graphical model* of the factors $\mathbf{z}$, such that each sample $\mathbf{z}_i$ is distributed according to $\mathcal{N}\left(\mathbf{0}, \mathbf{\Lambda}^{-1}\right)$.

5.1.1 Low-rank plus sparse-inverse covariance

Given a dataset $\mathbf{Y} \in \mathbb{R}^{n \times p}$, our goal is to infer the sparse structure of the underlying GMRF, encoded by the sparse-inverse covariance term $\mathbf{\Lambda}^{-1}$. If $\mathbf{y}_i$ is truly sampled from a Gaussian with sparse precision $\mathbf{\Lambda}$, then we can efficiently estimate $\mathbf{\Lambda}$ with the *graphical-Lasso* algorithm [GLasso, Banerjee *et al.*, 2008; Friedman *et al.*, 2008]. The challenge is to estimate $\mathbf{\Lambda}$ *in the presence of low-rank structures* (in the marginal covariance), induced by *confounding* latent variables $\mathbf{X} \in \mathbb{R}^{n \times q}$. We show that a low-rank structure leads to *highly correlated* covariates, which in turn increases the number of false edges called by GLasso (on the empirical covariance of $\mathbf{Y}$). Our approach is to perform GLasso on the sample-covariance of $\mathbf{Z} \in \mathbb{R}^{n \times p}$, the *unconfounded* version $\mathbf{Y}$.

[1] Using the sub-index notation here as a descriptor.

Based on the discussion above, we build the following generative model:

$$\begin{aligned} \mathbf{y}|\mathbf{x},\mathbf{z} &\sim \mathcal{N}\left(\mathbf{W}\mathbf{x}+\mathbf{z}, \sigma^2\mathbf{I}\right) \\ \mathbf{x} &\sim \mathcal{N}\left(\mathbf{0},\mathbf{I}\right) \\ \mathbf{z} &\sim \mathcal{N}\left(\mathbf{0},\boldsymbol{\Lambda}^{-1}\right) \\ p(\boldsymbol{\Lambda}) &\propto \exp\left(-\lambda||\boldsymbol{\Lambda}||_1\right)\ , \end{aligned} \tag{5.3}$$

where $\boldsymbol{\Lambda}$ is sampled from a Laplace distribution (a sparsity promoting prior) and the level of sparsity is driven by the hyperparameter (or regularisation parameter) λ. Figure 5.1 shows the corresponding graphical model. We propose a hybrid approach of EM and RCA to optimise this generative model with respect to the loadings $\mathbf{W}$ and the sparse GMRF encoded by the precision matrix $\boldsymbol{\Lambda}$.

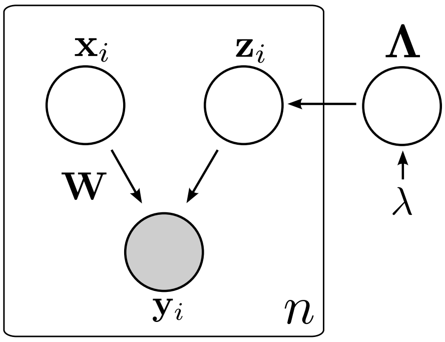


Figure 5.1: Generative model that yields a low-rank plus sparse-inverse structure in the marginal covariance. The parameters are optimised by an EM/RCA hybrid.

Averaging over $\mathbf{X} \in \mathbb{R}^{n \times q}$ in the graphical model yields the joint log-density

$$\ln p(\mathbf{Y},\boldsymbol{\Lambda}\,|\,\mathbf{W}) = \sum_i \ln\left\{\mathcal{N}\left(\mathbf{y}_i\,|\,\mathbf{0},\mathbf{W}\mathbf{W}^\top+\boldsymbol{\Lambda}^{-1}\right)p\left(\boldsymbol{\Lambda}\right)\right\} \tag{5.4}$$

$$\geq \int q(\mathbf{Z}) \ln \frac{p(\mathbf{Y},\mathbf{Z},\boldsymbol{\Lambda})}{q(\mathbf{Z})}\,\mathrm{d}\mathbf{Z}\ . \tag{5.5}$$

The integral in eq. (5.5) acts as a *variational lower bound*[1] of the joint log-density in eq. (5.4) and $q(\mathbf{Z})$ is the *variational* distribution that we must optimise to raise the bound. Because the parameters $\mathbf{W}$ and $\boldsymbol{\Lambda}$ have no fixed-

[1] Computation is reduced to optimising a functional with respect to the distribution $q()$.

point solution, we seek a MAP solution by optimising the lower bound via the expectation-maximisation algorithm [EM, Dempster *et al.*, 1977; Lawrence *et al.*, 2010]. We derive the variational lower bound and equations for updates in appendix A.4.

E-step Replacing $q(\mathbf{Z})$ with the posterior $p\left(\mathbf{Z}\,|\,\mathbf{Y}, \widehat{\boldsymbol{\Lambda}}\right)$ for current estimates $\widehat{\boldsymbol{\Lambda}}$ and $\widehat{\mathbf{W}}$, amounts to the following E-step for the *exact* update of the posterior distribution of $\mathbf{z}_i\,|\,\mathbf{y}_i$:

$$\operatorname{var}[\mathbf{z}\,|\,\mathbf{y}] = \left(\left(\widehat{\mathbf{W}}\widehat{\mathbf{W}}^\top + \sigma^2\mathbf{I}\right)^{-1} + \widehat{\boldsymbol{\Lambda}}\right)^{-1} \tag{5.6}$$

$$\mathbb{E}[\mathbf{z}_i\,|\,\mathbf{y}_i] = \operatorname{var}[\mathbf{z}\,|\,\mathbf{y}]\left(\widehat{\mathbf{W}}\widehat{\mathbf{W}}^\top + \sigma^2\mathbf{I}\right)^{-1}\mathbf{y}_i \tag{5.7}$$

$$\mathbb{E}_{p(\mathbf{z}\,|\,\mathbf{y})}[\mathbf{z}_i\mathbf{z}_i^\top] = \operatorname{var}[\mathbf{z}\,|\,\mathbf{y}] \;+\; \mathbb{E}[\mathbf{z}_i\,|\,\mathbf{y}_i]\,\mathbb{E}[\mathbf{z}_i\,|\,\mathbf{y}_i]^\top \;. \tag{5.8}$$

M-step Then for fixed $\widehat{\mathbf{Z}}$, the only free parameter in the expected complete-data log-likelihood $\mathcal{Q} = \mathbb{E}_{p(\mathbf{Z}\,|\,\mathbf{Y})}\left[\ln p\left(\mathbf{Z}, \boldsymbol{\Lambda}\right)\right]$ is the sparse-inverse $\boldsymbol{\Lambda}$. Therefore, the update for $\boldsymbol{\Lambda}$ depends on the L1 problem:

$$\widehat{\boldsymbol{\Lambda}} \quad = \quad \max_{\boldsymbol{\Lambda}} \quad \ln|\boldsymbol{\Lambda}| - \operatorname{tr}\left(\tfrac{1}{n}\sum_i \left\{\mathbb{E}_{p(\mathbf{z}\,|\,\mathbf{y})}[\mathbf{z}_i\mathbf{z}_i^\top]\right\}\boldsymbol{\Lambda}\right) - \lambda||\boldsymbol{\Lambda}||_1 \;, \tag{5.9}$$

which can be maximised efficiently with the *graphical-Lasso* algorithm [GLasso, Banerjee *et al.*, 2008; Friedman *et al.*, 2008] (note the L1-penalty for sparsity).

RCA-step After one iteration of EM, we update $\widehat{\mathbf{W}}$ via RCA based on the newly estimated $\widehat{\boldsymbol{\Lambda}}$, by first solving for $\mathbf{S}$ in the GEP:

$$\begin{aligned} \tfrac{1}{n}\mathbf{Y}^\top\mathbf{Y}\mathbf{S} &= \boldsymbol{\Sigma}_{GMRF}\mathbf{S}\mathbf{D} \quad \text{where} \quad \boldsymbol{\Sigma}_{GMRF} = \boldsymbol{\Lambda}^{-1} \\ \widehat{\mathbf{W}} &= \boldsymbol{\Sigma}_{GMRF}\mathbf{S}(\mathbf{D}-\mathbf{I})^{1/2} \;. \end{aligned} \tag{5.10}$$

Algorithm 2 summarises the EM and RCA steps which collectively constitute one iteration of the EM/RCA hybrid:

Algorithm 2 EM/RCA

```
Initialise $\sigma^2$ , $\widehat{\mathbf{W}}$ and $\widehat{\mathbf{\Lambda}}$ and $\lambda$ .
repeat
    E-step: Update posterior distribution of $\mathbf{Z}|\mathbf{Y}$ with (5.6) and (5.7).
    M-step: Update $\widehat{\mathbf{\Lambda}}$ with (5.9).
    RCA-step: Update $\widehat{\mathbf{W}}$ with (5.10).
until the lower-bound (5.5) converges.
```

Related work

A more generalised approach was proposed recently by Agakov *et al.* [2012], where the sparsity assumption is on the *joint* field of observed and latent variables. Under this framework, our "low-rank plus sparse-inverse" approach becomes the special case where there are as many latent variables as there are observed (a one-to-one correspondence) and we focus only on structure learning of the latent field. The authors also consider straightforward extensions to discriminative mixtures of such fields, where each expert is activated based on side information, and for "Gaussianising" long-tailed marginals through Gaussian copulas.

Another closely related approach was by Chandrasekaran *et al.* [2010] where the marginal precision matrix of the observed variables is decomposed into a sum of *sparse plus low-rank* terms. This occurs when the latent dimensionality is smaller than the observed and the *conditional* precision matrix (of the observed given the latents) is assumed to be sparse. Then the sparse/low-rank decomposition naturally appears as the Schur complement of the latent variables' (lower-right) block in the joint precision matrix (see also Appendix A.1).

5.1.2 Experiments

For each experiment, we initialise:

- the noise variance as $\sigma^2 = \frac{1}{2p} \operatorname{tr} \widehat{\mathbf{S}}$, where $\widehat{\mathbf{S}}$ is the sample-covariance of the data $\mathbf{Y} \in \mathbb{R}^{n \times p}$. Note that if we fix σ^2 to the initialized value, this implicitly fixes the number of latent variables (confounders). A more systematic approach would be a line search on σ^2 during the M-step or using the BIC criterion over a small range of q (number of latent variables);

- the loadings matrix as $\mathbf{W} = \mathbf{U}_q(\mathbf{L}_q - \sigma^2\mathbf{I})^{1/2}$ with the q principal eigenvectors in $\mathbf{U}_q$ whose eigenvalues are larger than σ^2;
- the sparse GMRF-encoding matrix as $\boldsymbol{\Lambda} = \mathbf{I}$ (no dependencies);
- a sequence of L1-regularisation parameters as $\lambda = 5^x$ such that x is linearly interpolated in [-8,3], thus creating a solution "path"[1] as λ increases exponentially. The solution paths of *lasso*-based algorithms tend to be unstable. Therefore, we apply a form of *stability selection* [Meinshausen & Bühlmann, 2010] to smoothen the solution paths: for each λ and method, the results are stabilised by taking 100 repeats with a random 90% sub-sampling for each repeat. If an edge of the GMRF is called (that is, estimated as non-zero in $\boldsymbol{\Lambda}$) on more than 50% of the repeats then it is declared *active*.

A result for a particular regularisation parameter λ constitutes an estimate $\widehat{\boldsymbol{\Lambda}}$. The estimate is compared to the ground-truth network, represented by the adjacency matrix $\mathbf{G}$. For some $\Lambda_{ij} \neq 0$, the call is *true-positive* (*TP*) if $G_{ij} \neq 0$ and *false-positive* (*FP*) if $G_{ij} = 0$. The efficiency of the algorithm is measured in terms of *recall* $= \frac{\#TP}{\#P}$ and *precision* $= \frac{\#TP}{\#TP\,+\,\#FP}$ where $\#P$ are the total true edges. As the λ parameter increases to the next number in the sequence, EM/RCA continues from the point where it last converged, thus tracing a performance path in the recall-precision space.

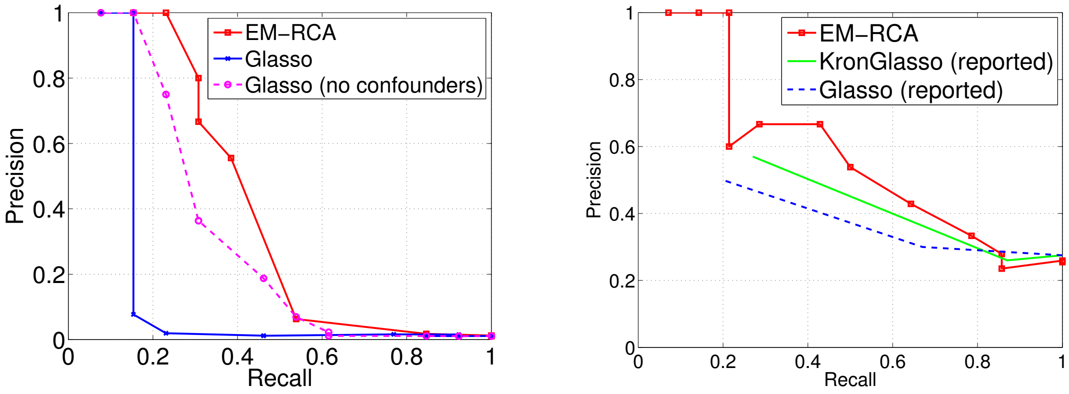
Simulations

We consider an artificial dataset sampled from the generative model in eq. (5.3), Figure 5.1, to demonstrate the effects that confounders have on the estimation of the sparse-inverse covariance. Specifically the raw data are generated as:

$\mathbf{Y} = \mathbf{X}\mathbf{W}^\top + \mathbf{Z} + \mathbf{E}$, where $\mathbf{Y}, \mathbf{E} \in \mathbb{R}^{100\times 50}$, $\mathbf{W} \in \mathbb{R}^{50\times 3}$, $\mathbf{X} \in \mathbb{R}^{100\times 3}$. For each sample, $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_3)$, $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda}^{-1})$, $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ and

[1] A path of sparse-inverse estimates, where the estimate for some λ_i is used as the initialisation for λ_{i+1}.

$\mathbf{w}_j \sim \mathcal{N}(\mathbf{0}, \gamma\mathbf{I}_{50})$. The sparsity of $\boldsymbol{\Lambda}$ is 1% of all $p(p-1)/2$ potential edges in the GMRF. The non-zero entries of $\boldsymbol{\Lambda}$ are iid samples from $\mathcal{N}(1, 2)$. The variance γ is such that $\boldsymbol{\Lambda}^{-1}$ and $\mathbf{W}\mathbf{W}^{\top}$ explain an equal amount of variance and the variance σ^2 of the induced noise is such that the signal-to-noise ratio (SNR) is 10. Figure 5.2(a) shows the effectiveness of EM/RCA on a dataset suffering from confounders, whereas standard GLasso fails to find any part of the true structure even when strongly regularised. The EM/RCA algorithm has significantly better performance than GLasso on the confounded data (solid curves). The dashed curve shows the performance of GLasso on the same samples but without confounders ($\mathbf{W}$ is zero). We note that EM/RCA on the confounded data performs better than GLasso on the unconfounded data because the latter have a lower SNR.



(a) Simulation

(b) Sachs

Figure 5.2: (a) Recall-precision curves of EM/RCA and GLasso on the simulated confounded dataset (solid curves), and GLasso on the simulated non-confounded dataset (dashed curve). (b) EM/RCA, KroneckerGLasso and GLasso on the Sachs data. The Kronecker-GLasso and GLasso curves are taken from [Stegle *et al.*, 2011].

Reconstructing a protein-signaling network

We compare EM/RCA to the *Kronecker*-GLasso algorithm of Stegle *et al.* [2011] on the protein-signaling data from [Sachs *et al.*, 2005]. These data provide signal measurements from $p = 11$ proteins under various external stimuli. We collect $n = 2,666$ samples from the first three experiments into one heterogeneous dataset $\mathbf{Y} \in \mathbb{R}^{n \times p}$. The heterogeneous conditions of these experiments induce confounding effects in the data. For the sake of comparison, we also run the analysis on a random 10% subset of the n samples, with a 10% from each of the three experiments. All results are validated based on the *moralised*[1] version of the directed ground-truth network, constructed and validated biologically by Sachs *et al.* [2005]. Figure 5.2(b) shows EM/RCA slightly outperforming the other methods and Figure 5.3 shows the reconstructed networks for recall 0.4. We observe that EM/RCA appears more conservative in calling positive edges

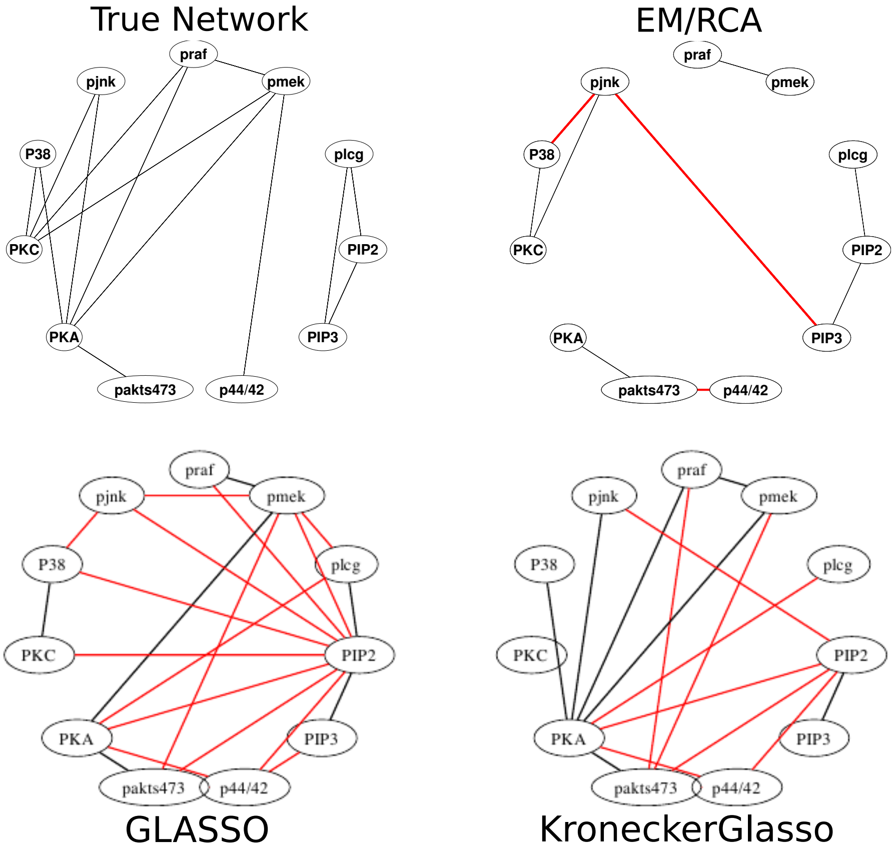


Figure 5.3: Reconstructed networks by EM/RCA, Kronecker-GLasso and GLasso on the Sachs data, for 0.4 recall. Red edges indicate false-positives.

[1] For any node, its parents are connected and all edges become undirected.

while preserving a higher precision.

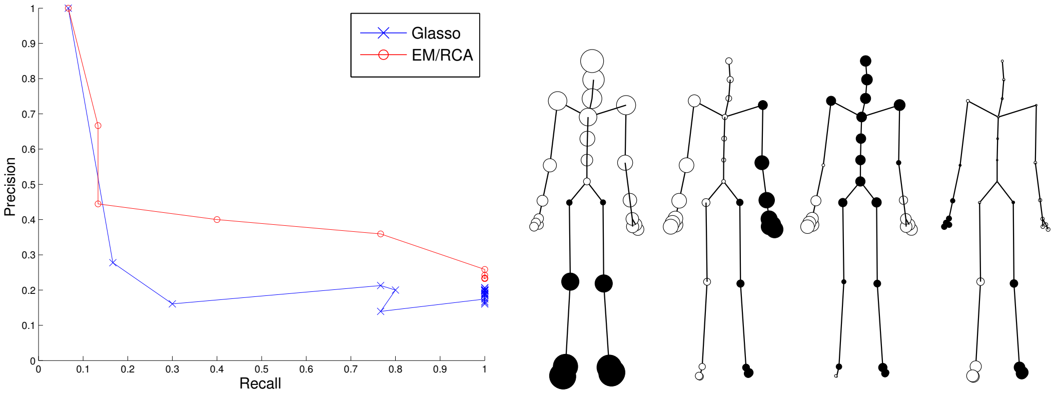
Reconstructing the human form

The data in this experiment come from the CMU motion-capture database[1]. The objective is to reconstruct the underlying connectivity of the human form, given only the 3D locations of 31 sensors placed about the figure's body. Each captured motion in the database involves data of the skeleton (or stickman) specific to the person under the trial (different heights, builts, etc.) and the 3-D sensor cloud data. Each trial involves 31 sensors, so the raw dataset for each trial has size n (frames captured in the trial) $\times$ 93 (x,y,z $\cdot$ sensors).

The aim of our model is to recover the connectivity between these sensors. This should be possible because we expect sensors that are connected in the underlying figure to be conditionally independent of other sensors in the figure. This motivates the underlying sparse structure. Conversely, different motions exhibit much broader correlations across the figure. In particular, walking exhibits anti-correlations between sensors on different legs and across the arms. These types of motion should be far better recovered through a low-rank representation of the covariance.

If, as expected, the raw data is confounded by low-rank properties associated with particular structured motions (as opposed to random poses, as might be adopted by a wooden artist's doll) then our combination of low-rank with sparse connectivity should outperform a model based purely on sparse connectivity. We therefore compare EM/RCA and GLasso on trials involving walking, running, jumping and dancing. The local connectivity between the sensors, i.e. the human skeleton, should be represented in the sparse matrix $\mathbf{\Lambda}$ (prescribing a Gaussian random field). To further motivate this idea we also note the physical interpretation of $\mathbf{\Lambda}$ as the stiffness (or Laplacian) matrix of a spring network, where the off-diagonal entries represent the negative stiffness of the spring. Therefore, to detect an "attracting" connection between two sensors we look only for negative

[1] http://mocap.cs.cmu.edu.



(a) (b)

Figure 5.4: (a) Recall-precision curves of EM/RCA and GLasso on the CMU motion-capture data. (b) Hinton diagram of $\mathbf{X}$ capturing the confounding effects in the motions. Each column of $\mathbf{X}$ is visualised by rearranging its elements to the corresponding sensors on the ground-truth stickman. The colour of a dot indicates the sign and the size is proportional to the magnitude of the corresponding element in $\mathbf{X}$.

entries in the estimated $\mathbf{\Lambda}$[1].

Data preprocessing Because we are interested in modeling interactions between sensors and to avoid modeling explicitly the correlations between spatial features (x-y-z coordinates) within a sensor, we convert absolute positions of the point cloud into inter-point distances. Hence the covariance to be analysed reduces from 93 features to 31 (number of sensors involved in a frame). Also, we treat the frames as independent, meaning that we ignore the sequence in which they appear in a trial. This amounts to summing up sensor-covariances across all frames. Let $\mathbf{H}(k) \triangleq \mathbf{I} - \frac{1}{k}\mathbf{1}\mathbf{1}^\top$ be the *centering* operator, where $\mathbf{1} \in \mathbb{R}^k$ is the vector of ones; $\mathbf{D}^{(f)}$ is the *squared distance* matrix for some configuration of points (sensors) $\mathbf{X}_{(f)} \in \mathbb{R}^{31\times 3}$ at frame f, such that

$$D_{ij} = (\mathbf{x}_i - \mathbf{x}_j)^\top(\mathbf{x}_i - \mathbf{x}_j) = \mathbf{x}_i^\top\mathbf{x}_i - 2\mathbf{x}_i^\top\mathbf{x}_j + \mathbf{x}_j^\top\mathbf{x}_j \ ,$$

[1] On a similar note, see also MacKay's "*The Humble Gaussian distribution*" on interpreting Gaussian graphical models as energy models.

and in matrix notation $\mathbf{D} = \mathbf{1}\,\mathrm{diag}\left(\mathbf{X}\mathbf{X}^\top\right)^\top - 2\mathbf{X}\mathbf{X}^\top + \mathrm{diag}\left(\mathbf{X}\mathbf{X}^\top\right)\mathbf{1}^\top$. Centering $\mathbf{D}$ gives the centered squared distance matrix

$$\mathbf{HDH} = -2\mathbf{H}\mathbf{X}\mathbf{X}^\top\mathbf{H} \propto \bar{\mathbf{X}}\bar{\mathbf{X}}^\top = \bar{\mathbf{K}}\ , \quad \text{since} \quad \mathbf{H}\mathbf{1} = \mathbf{0}\ . \tag{5.11}$$

Hence computing the centered squared distance matrix is equivalent to computing the centered inner-product matrix (that is, the inner-product matrix of the centered raw data $\bar{\mathbf{X}}$). Let $\bar{\mathbf{x}}_j \in \mathbb{R}^{31}$ denote the j-th column of $\bar{\mathbf{X}}$, collecting the x-only-coordinates of all 31 sensors (or y, z depending on j) for a particular frame. Then from eq. (5.11), the sum across frames $\sum_f \bar{\mathbf{X}}_{(f)}\bar{\mathbf{X}}_{(f)}^\top = \sum_f \bar{\mathbf{K}}_{(f)}$ can be seen as a sum of independent scatter matrices in the dual representation, and since $\bar{\mathbf{X}}\bar{\mathbf{X}}^\top = \sum_j \bar{\mathbf{x}}_j\bar{\mathbf{x}}_j^\top$ then $\sum_f \bar{\mathbf{K}}_{(f)} = \sum_{f,j} \bar{\mathbf{x}}_j^{(f)}\bar{\mathbf{x}}_j^{(f)\top}$, which is the scatter matrix of all $\bar{\mathbf{x}}_j$ vectors as samples (rows) in our final design matrix $\mathbf{Y}$. To summarise the preprocessing, we start with raw data $\mathbf{X}_{(f)} \in \mathbb{R}^{31\times 3}$ for each frame $f \in \{1, .., F\}$, center its rows via $\mathbf{H}\mathbf{X}_{(f)} = \bar{\mathbf{X}}_{(f)}$ and collect all frames in the design matrix $\mathbf{Y}^\top = [\bar{\mathbf{X}}_{(1)} \ldots \bar{\mathbf{X}}_{(F)}]$. This amounts to working with the dual representation of the data and treating as independent the frames as well as the sensor (x,y,z) coordinates.

Results Figure 5.4 shows the recall-precision curves for GLasso and EM/RCA on the CMU mocap data, where EM/RCA consistently outperforms standard GLasso. Figure 5.5 shows the stickmen recovered by EM/RCA and GLasso. We note that the connectivities and eigenposes are more faithful to the true human form, in comparison to GLasso. For a small λ setting (recall 1) the precisions are similar; nonetheless the human form is robust, with very weak (yellow) edges wherever they do not apply (e.g. elbow-waist, elbow-head). This signifies that the precision measure might be ill-suited for evaluating a stickman, where the network configuration has a spatial interpretation. The ground-truth is also "noisy" in the sense that a shoulder-chest edge, for instance, must be called as the torso is a rigid part of the human body (high stiffness)

Figure 5.4(b) illustrates the confounding effects captured by $\mathbf{X}$ (as part of $\mathbf{X}\mathbf{X}^\top + \boldsymbol{\Lambda}^{-1}$). Specifically, in the first component, the legs are anti-correlated to

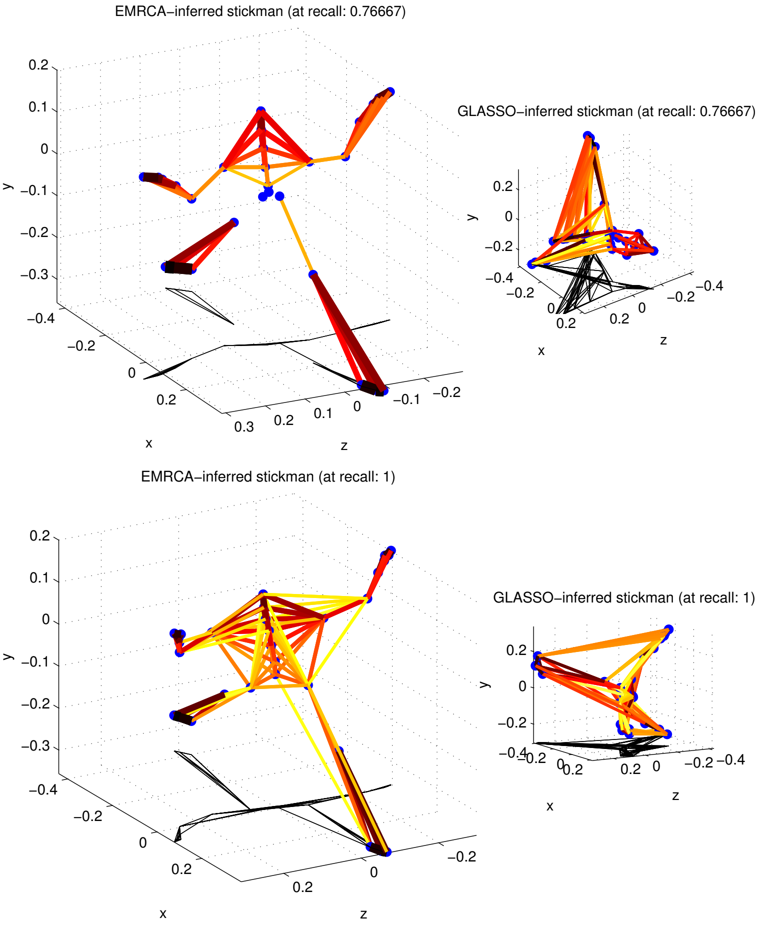


Figure 5.5: Stickman reconstructions by EM/RCA (left) and GLasso (right) for recalls 0.77 (top) and 1 (bottom). For each stickman, inferred edges are superimposed on the *eigenposes* extracted from the 3 principal eigenvectors of the estimated sparse $\mathbf{\Lambda}$ (or Laplacian of the spring system). Edge color and thickness indicate the negative stiffness intensity (red is large) and the black lines are shadows for aiding the perspective.

the upper-half of the body, which can be attributed to jumping motions. The second and forth components capture anti-correlations across the different legs and arms, exhibited by walking and running, as discussed earlier. Finally, the third component shows strong anti-correlation between the hands and the rest of the upper-body, which is more open to interpretation.

Discovering collusion patterns in voting data

And now for something completely different[1]. In this section we analyse voting data from the Eurovision song contest collected[2] across recent years. The residents of each country vote for the best song (other than its own). Each country thus produces a ranking which is translated into points; 12, 10, 8, 7, 6, 5, 4, 3, 2, 1 for the top ten. The country with the most points wins the song contest.

More precisely, each sample (row) $\mathbf{y}_i$ in our design matrix consists of the votes that a particular country gave to every other country (from a complete alphabetically ordered list of countries) in a particular year of the competition, and it has the following format: (`# votes to Albania`, `#votes to Andorra`, ..., `#votes to United Kingdom`). We assume that any country always rewards the maximum allowed points to itself (a country always likes its own song), and the whole row forms an affinity vector of the country towards all countries (including itself) for the duration of one competition.

The goal here is to reconstruct a network of collusive voting, that is, determine the pairs of countries that tend to vote on any basis of factors other than song quality/popularity (for instance, political relations, geography, etc). Naturally, we assume this network to be sparse and we relax the ordinal (non-Gaussian) restriction of the variables such that they follow a Gaussian graphical model. We also expect the network to form geographically relevant clusters.

As in the previous experiment, we are interested in positive interactions (col-

[1] This experiment was inspired from Martin O'Leary's blog-post: `http://mewo2.com/nerdery/2012/05/20/ive-got-eurosong-fever-ted/`.

[2] An early version of this dataset was compiled by Anthony Goldbloom of Kaggle, the extended version published by Martin O'Leary at `https://github.com/mewo2/eurovision/`.

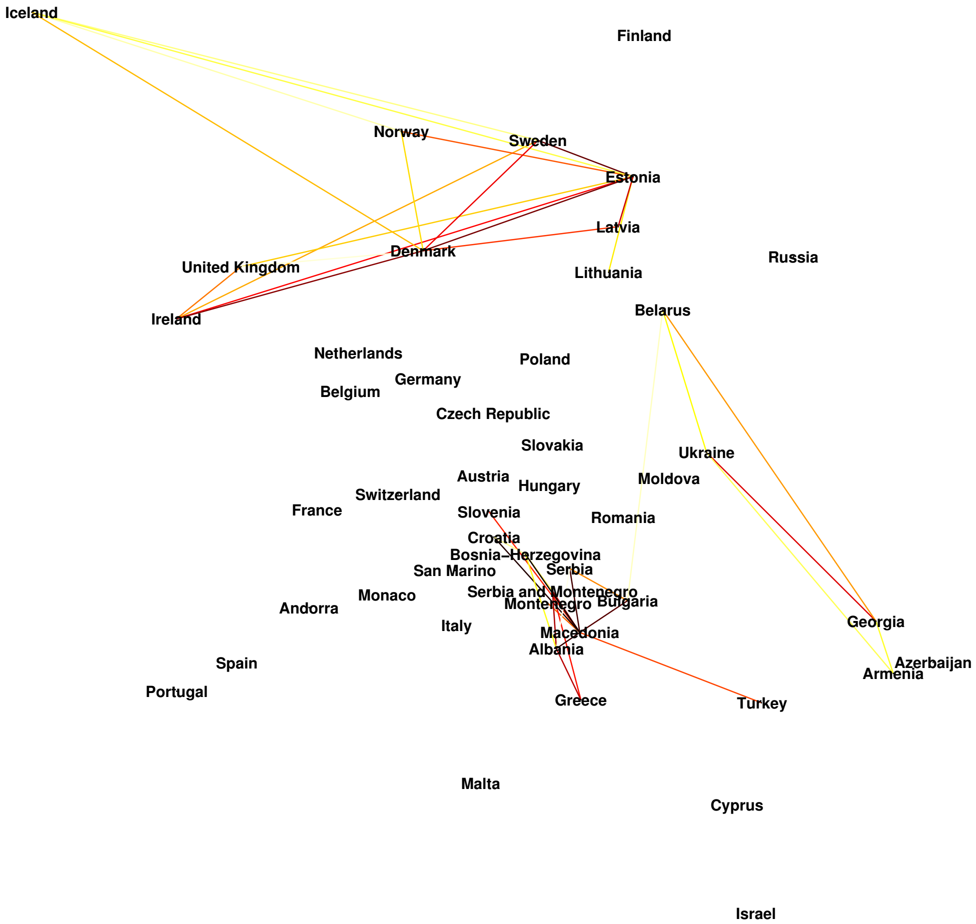


Figure 5.6: The emergence of voting blocs in the Eurovision song contest. Only edges of negative entries in the precision are shown, which imply “attracting” links. Three distinct blocs are visible: the northern bloc consisting mainly of Britannic, Scandinavian and Baltic states, the eastern bloc consisting only of post-Soviet states, and the southern bloc consisting mainly of Balkan and Slavic states. Darker edges imply stronger conditional dependencies. The coordinates forming this map are artificially induced and not part of the output.

lusions) so we focus on the negative entries of the estimated precision. Unfortunately this case has no ground truth; nonetheless, there is some room for qualitative evaluation: namely, the topology of the collusion network in Figure 5.6 reflects to some extent the European geography (that is, most conditional dependencies are restricted between geographical neighbors).

5.2 Analysing residuals of a Gaussian process

Differences in gene expression profiles We now revisit the analysis of gene expression time-series from chapter 2. To reiterate the problem, a common challenge in data analysis is to summarize the difference between treatment and control samples. To illustrate how RCA can help, we consider two gene expression time-series of cell lines. The treatment cells are targeted by TP63 introduced into the nucleus by tamoxifen. The control cells are simply subject to tamoxifen alone. The data used for this case study come from [Della Gatta *et al.*, 2008][1]. The treatment group $\mathbf{Y}_1 \in \mathbb{R}^{n_1 \times p}$ contains n=13 time-points of $p = 22,690$ gene expression measurements, whilst the control group $\mathbf{Y}_2 \in \mathbb{R}^{n_2 \times p}$ contains only $n_2 = 7$ time-points. This complexity of data (with different numbers of time-points and non-uniform sampling) is typical of many bio-medical data sets. The challenge is to represent the differences between the gene expression profiles for these two data sets. CCA could be applied but this would represent the similarities between the data, not the differences.

First we consider the null hypothesis that both time-series are identical. This implies that $\mathbf{y}^\top = (\mathbf{y}_1^\top \; \mathbf{y}_2^\top)$ can be modeled by a Gaussian process (GP) with a temporal covariance function, $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$, where $\mathbf{K} \in \mathbb{R}^{n \times n}$ for $n{=}n_1{+}n_2$ is structured such that both $\mathbf{y}_1$ and $\mathbf{y}_2$ are generated from the same function, $K_{i,j} = k(t_i, t_j) = \exp(-\frac{1}{2}\ell^{-2}(t_i - t_j)^2)$, a squared-exponential covariance function (or RBF kernel, figure 5.7(a)). Other kernels could be used and the hyperparameters of the kernel could be optimized, but for this simple demonstration we set $\ell = 20$ which provides a bandwidth roughly in line with the time-point sampling

[1] GEO database, accession number GSE10562.

intervals. We also add a small noise term along the diagonal of $\mathbf{K}$ which was set to 1% of the data variance.

Now by the null hypothesis assumption, a more general model (dual paradigm) of the form $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{X}\mathbf{X}^\top + \mathbf{K})$, should explain no variance in the low-rank component $\mathbf{X}\mathbf{X}^\top$, as all the signal in the time-series is assumed to be explained by the underlying function sampled from the GP. If we solve for the residual components $\mathbf{X}$ via RCA, they will be forced to explain how the two time-series are actually different.

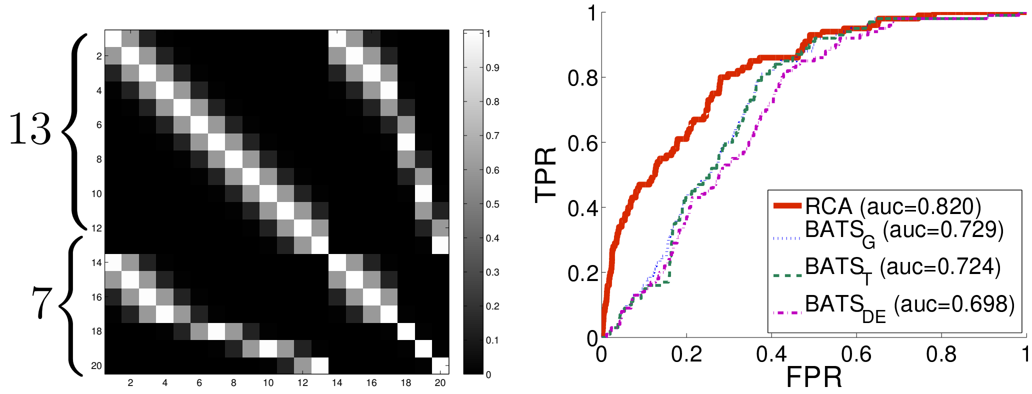
We project the profiles onto the eigen-basis of the first q generalised eigenvectors $\tilde{\mathbf{Y}} = \mathbf{S}_q^\top \mathbf{Y}$ and obtain a score of differential expression based on the norms of their projections. The number q of retained principal eigenvectors is decided on the number of corresponding eigenvalues d_i being larger than one. Recall from PPCA, that as we increase the assumed noise variance σ^2, more eigenvalues become negative and less eigenvectors are retained in $\widehat{\mathbf{W}}$ of eq. (3.3), p. 39. Similarly, RCA standardises any noise in eq. (3.19), so we only have to retain the eigenvectors of eigenvalues larger than 1. In this case, the assumed noise variance embedded in the kernel drives the effective number of eigenvectors in the projection basis.

We rank the scores and compare to the noisy ground-truth list of binding targets[1] of TP63 from [Della Gatta *et al.*, 2008], giving the ROC performance curve in Figure 5.7(b). The baseline method that we compare against is the Bayesian hierarchical model BATS [Angelini *et al.*, 2007]. Note that RCA outperforms BATS in terms the area under the ROC curve for all of its noise models.

5.3 Summary

Full covariance matrix models of data are often problematic as their parameterisation scales with p^2. Two separate approaches to a reduced parameterization of these matrices are to base them on low-rank matrices (as in probabilistic PCA)

[1] A gene with a large number of binding sites for TP63 is a strong candidate for being one of its direct targets, and thus associated with TP63-related diseases.



(a) (b)

Figure 5.7: (a) An RBF covariance computed on the augmented time-input vector for the microarray experiment. The covariance is computed across the times $\mathbf{t}^\top = (0 : 20 : 240, 0, 20, 40, 60, 120, 180, 240)$ jointly for control and treatment. (b) ROC comparison against BATS variants of different noise models (G: Gaussian, T: t-distribution, DE: double-exponential). See also [Kalaitzis & Lawrence, 2011b] for an alternative approach based on GPs.

or on a sparse-inverse structure (as in GLasso). These two approaches have very different characteristics: one assumes that a reduced set of latent variables is governing the data, the other involves specifying sparse conditional dependencies in the data. More precisely, when the data marginal is Gaussian, the precision (inverse-covariance) matrix induces a Gaussian Markov random field. Furthermore, a *sparse* precision possesses a valuable graphical interpretation, put to fruition as an efficient regression model or as a structure learning approach. Clearly, in any given dataset both low-rank and sparse-GMRF characteristics may be present.

After describing the RCA framework in chapter 3, in the present chapter we addressed the above problem to motivate the particular case of the explained covariance term $\mathbf{\Sigma}$ being sparse-inverse. We proposed a basic point-estimation algorithm based on EM for learning the low-rank and sparse-inverse parts of the marginal covariance. This was demonstrated to good effect with experiments spanning computational biology, with an example of a small protein-signaling network; motion-capture, where the results became much more visually interpretable; and a “socio-political” example, where we showed evidence of collusion in the Eurovision voting “system” amongst participating nations.

As an attempt to tie RCA to GPs from Chapter 2, we closed with a simple demonstration of explaining away the trained covariance of a GP (defined by a RBF) on concatenated time-series from two different conditions. The residual structure served as the basis for measuring the differential expression across the experiments.

Armed with a background on sparse-inverse selection, for the next and final chapter we will focus solely on the sparsity of precision matrices of matrix-normal (or matrix-Gaussian) models, that is, Gaussian densities over *random matrices*. We will use this distribution on design matrices and learn its two precision matrix parameters. One is the precision over the rows of a matrix-sample and the other is precision over the columns. We will show that to simultaneously learn the structure of those two graphs is at least as hard as an iterative application of GLasso, which is provably efficient in itself. We will motivate the Kronecker-sum as a novel structure for the joint precision of matrix-normal, which borrows from Algebraic Graph Theory and provides an easily interpretable factorisation of the precision, among other benefits.

Chapter 6

The Bigraphical Lasso

Until now, we have embraced the endemic assumption in machine learning of i.i.d. data. We now look at the more general case where this assumption can be flawed: more complex data sets can exhibit partial correlations between data points as well as features. To deal with correlation of this type we introduce the *bigraphical Lasso*. The model is based on a Gaussian distribution over random matrices that specifies correlations between data points and features. It does so with a structured (*Kronecker-sum*) precision matrix that induces a Cartesian product of undirected graphs, a prominent product with well studied properties in spectral graph theory. One factor represents the graph over the rows of the matrix and the other the graph over the columns. This structure has appealing properties for regression and enhanced interpretability.

The most general of such matrix-models has a number of parameters that scales quadratically with features and data points. To deal with this parameter explosion we introduce ℓ_1 penalties and fit the model through a flip-flop algorithm that reduces the problem to a series of lasso regressions. We demonstrate the performance of our approach with extensive simulations and an example from the COIL image data set.

6.1 Introduction

When fitting Gaussian models to data, we usually make an independence assumption across data points and fit the covariance matrix by maximum likelihood. The number of parameters in the covariance matrix can be reduced by factor analysis like structures (see Chapters 3 and 4) or by constraining the inverse-covariance (or precision matrix) to be sparse [e.g. Banerjee *et al.*, 2008]. A sparse precision matrix defines a Gaussian Markov random field which is conveniently represented by a weighted undirected graph. Nodes which are not neighbors in the graph are conditionally independent given all other nodes. Models specified in this way can learn conditional independence structures between features.

An alternative Gaussian modeling approach was introduced by Lawrence [2012], who showed that spectral dimensionality reduction methods have an interpretation as sparse graphical models where the independence assumption is across data features, and the parameters of the covariance are fitted by maximum likelihood (or in the case of local linear embeddings [Roweis & Saul, 2000] by maximizing a pseudolikelihood). This assumption leads to much better determined parameters in the case where the number of features is greater than the number of data points (the so called large p, small n case).

The choice of feature independence or data point independence is a model choice issue, but both choices are in fact a simplification of a more general framework that aims to estimate the conditional independence relationships between both features and data points. It is this type of model that we address in this chapter. Specifically we want to build a sparse graph that interrelates both features and data points. For instance, we might have a data set that is a video. Here the data points are the frames of the video and the data features are the pixels in the video. Let's assume that the ordering of the video frames and the neighborhood structure between pixels has somehow been lost. A potential learning task would be to learn both the temporal structure of the data and the spatial structure of the inter related pixels. We successfully solve this task for a simple video from the COIL data set in Section 6.5.

An alternative motivating data example could be gene expression data, where we might wish to extract a genetic network from the gene expression values whilst explaining correlations between samples (such as close genetic relationships, or related experiments) with a separate network. Econometrics, computational biology and computer vision are very few example domains that often deal with datasets of complex dependency structures that are best approximated with higher-dimensional models of matrices or tensors. Such data are more naturally represented by *matrix-variate* distributions.

6.1.1 Graphical Lasso and the matrix-variate Gaussian

The *graphical lasso* [GLasso, Banerjee *et al.*, 2008; Friedman *et al.*, 2008] is a computationally efficient penalised likelihood algorithm for learning sparse structures of conditional dependencies or Gaussian Markov random fields (GMRF) over features of iid vector-variate Gaussian samples [Lauritzen, 1996].

The matrix-variate normal [Dawid, 1981; Gupta & Nagar, 1999] is a Gaussian density which can be applied to a matrix through first taking a vectorized (*vec*) representation[1] of the matrix samples $\mathbf{X} \in \mathbb{R}^{n \times p}$ and assuming the covariance matrix has the form of a Kronecker product between two covariance matrices, separately associated with the rows and columns of the data. The Kronecker product assumption for the covariance implies that the precision matrix is also a Kronecker product, which is formed from the Kronecker product of the precision matrices associated with the rows and columns ($\boldsymbol{\Psi} \otimes \boldsymbol{\Theta}$).

One approach to applying sparse graphical models to matrix data is to combine the Kronecker product structured matrix variate normal with the graphical Lasso. Dutilleul [MLE, 1999] used a flip-flop approach for maximum likelihood estimation of the parameters of the matrix-normal and much later Zhang & Schneider [2010] used it for MAP estimation with sparsity penalties on the precision matrices. More recently, Leng & Tang [2012] applied the SCAD penalty [Fan & Li, 2001] as well as the Lasso in the likelihood function of the matrix-normal. Tsiligkaridis *et al.* [2013] analyzed the convergence of Kronecker GLasso under

[1] Vectorization of a matrix involves converting the matrix to a vector by stacking the columns of the matrix.

asymptotic conditions as well as simulations that show significant convergence speedups over GLasso and MLE.

However, whilst the Kronecker-product structure arises naturally when considering matrix-normals (Kronecker-normals), it is relatively dense when it comes to the dependencies it suggests between the rows. More precisely, if Ψ_{ij} in $\mathbf{\Psi} \otimes \mathbf{\Theta}$ is non-zero (for example, corresponding to an edge between samples i and j in the design matrix $\mathbf{X}$) then many edges between features of sample i and sample j (as many as in $\mathbf{\Theta}$) will also be active. A *sparser* structure would benefit situations where the connection between a feature of some sample and a different feature of any other sample is of no interest or redundant, simply because a same-feature dependency between different samples would suffice to establish a cross-sample dependency. For instance in a video, it is reasonable to assume given that the neighbors of pixel (i, j) in frame k are conditionally independent to the neighbors of pixel (i, j) in frame $k + 1$, conditioned on pixels (i, j) of both frames.

6.1.2 The Bigraphical Lasso

In this chapter, we introduce the *bigraphical Lasso* (BiGLasso), a model for matrix-variate data that preserves their column/row structure and, like the Kronecker product based matrix-normal, simultaneously learns two graphs, one over rows and one over columns of the matrix samples. The model is trained in a flip-flop fashion, so the number of Lasso regressions reduces to $\mathcal{O}(n + p)$. However, the model preserves the matrix structure by using a novel Kronecker *sum* structure for the precision matrix, $(\mathbf{\Psi} \otimes \mathbf{I}) + (\mathbf{I} \otimes \mathbf{\Theta})$ instead of the Kronecker product $(\mathbf{\Psi} \otimes \mathbf{\Theta})$. This structure enjoys *enhanced sparsity* in comparison to the conventional Kronecker-product structure of matrix-normals.

In the context of regression models, the Kronecker-sum prevents the conditional independence between responses of multi-output Gaussian processes, a property known in various literatures as *cancellation of inter-task transfer* or *autokrigeability*.

When operating on adjacency matrices, the Kronecker-sum is also known in algebraic graph theory as the Cartesian product of graphs and is arguably the most prominent of graph products [Chung, 1996; Imrich *et al.*, 2008; Sabidussi, 1959]. This endows the output of the BiGLasso with a more intuitive and interpretable graph decomposition of the induced Gaussian random field (GRF), see figure 6.1.

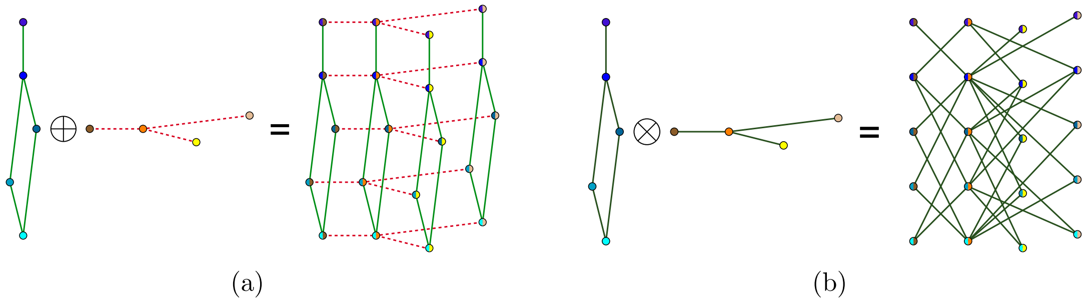


Figure 6.1: When acting on adjacency matrices of graphs, the Kronecker-sum acts as the Cartesian-product (a) and the Kronecker-product as the tensor-product (b). The lattice-like structure of the Cartesian-product is ideal for modeling dependencies between features as well as samples. More generally, since the Cartesian-product is associative, it can be generalized to model higher-dimensional GRFs. Note that here we do not include self-edges (zeros on the diagonals). Based on figures created by David Eppstein, http://en.wikipedia.org/wiki/Graph_product.

Enhanced Sparsity For a matrix density $\lambda \in [0,1]$ of both precision matrices the Kronecker-sum has $\mathcal{O}(\lambda np(n+p))$ non-zeros, whereas the Kronecker-product has $\mathcal{O}(\lambda n^2p^2)$ non-zeros.

Better Information Transfer Kronecker product forms have a known weakness, referred to in the Gaussian process (GP) literature as the cancellation of *inter-task transfer*: Bonilla *et al.* [2008, §2.3] showed that the predictive mean of a multi-output GP with a noise-free Kronecker-product covariance[1] and the same inputs conditioned across tasks (a conditioning structure referred to as a *block design*) uncouples the outputs of the different tasks, that is, the posterior factorises

[1] One factor for inter-task one for inter-point covariances.

and thus the outputs are computed independently. The key of this proof lies in the factorisable property of the inverse Kronecker-product, $(\mathbf{\Psi} \otimes \mathbf{\Theta})^{-1} = \mathbf{\Psi}^{-1} \otimes \mathbf{\Theta}^{-1}$. This property does not apply under the presence of additive noise, hence the outputs remain coupled. This result first arose in geostatistics under the name of *autokrigeability* [Wackernagel, 2003] and is also discussed for covariance functions by O'Hagan [1998]. Zellner [1962], Binkley & Nelson [1988] pointed out how the consideration of the correlation between regression equations leads to a gain in efficiency.

In a similar vein from econometrics, are models of *seemingly unrelated regressions* [SUR, Zellner, 1962], a form of *general* least squares that allows for a different set of regressors for each response. The problem reduces to *ordinary* least squares (OLS) when the same covariates are used across the outputs (block design). With a block design, OLS would pass on a potential gain in efficiency by disregarding correlations between responses. Nonetheless, the distribution of the maximum-likelihood estimators does not factorize, regardless of conditioning design. In contrast to SUR, a block design on a multi-output GP with a noise-free Kronecker-product covariance induces the stronger effect of conditional independence over the outputs. These two factorisations are very different and in general do not coincide.

The same property that allows for a simple flip-flop approach also negates the merit of exploiting any correlations between different outputs, but by coupling them with additive noise to enable inter-task transfer, flip-flop is no longer straightforward. Stegle *et al.* [2011] addressed this issue by adding iid noise to a Kronecker-product covariance — a *low-rank* factor for confounding effects and a *sparse-inverse* factor for inter-sample dependencies — and exploiting identities of the vec(.) notation for efficient computation within the matrix-normal model.

To summarize our contributions, contrary to existing approaches that use the Kronecker-product structure, the Kronecker-sum *preserves* the inter-task transfer. Our algorithm maintains the simplicity of the flip-flop with a simple trick of transposing the matrix-variate (samples become features and vice versa). At the same time, the induced Cartesian factorization of graphs provides a more parsimonious interpretation of the induced Markov network.

The rest of this chapter is structured as follows. We describe the matrix-normal model with the Kronecker-sum inverse-covariance in §6.2. In §6.3, we present the BiGLasso algorithm for learning the parameters of the Kronecker-sum inverse-covariance. We present some simulations in comparison to a recent Kronecker-normal model of Leng & Tang [SMGM, 2012] in §6.4 and an application to an example from the COIL dataset in §6.5. We conclude in §6.6.

6.2 Matrix-normal with the Kronecker-sum structure

To motivate our model, consider the case where matrix-variate data $\mathbf{Y}$ are sampled iid from a matrix-normal distribution (matrix-Gaussian). This is a natural generalisation of the Gaussian distribution towards tensor support[1]. This distribution can be reparametrized such that the support is over vectorised representations of random matrices,

$$\mathrm{vec}(\mathbf{Y}) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{\Psi}_n^{-1} \otimes \mathbf{\Theta}_p^{-1}\right).$$

The Kronecker-product-based SMGM Under the assumption that $\mathbf{\Psi}_n \otimes \mathbf{\Theta}_p$ is sparse, the SMGM estimator (Sparse Matrix Graphical Model) of Leng & Tang [2012] for the precision matrices $\mathbf{\Psi}_n, \mathbf{\Theta}_p$ can be computed iteratively by minimizing a flip-flop extension of GLasso for Kronecker-product matrix-normals using the ℓ_1 penalty:

$$\min_{\mathbf{\Theta}_p, \mathbf{\Psi}_n} \left\{ \frac{1}{Nnp} \sum_{i=1}^{N} \mathrm{tr}\left(\mathbf{Y}_i \mathbf{\Theta}_p \mathbf{Y}_i^\top \mathbf{\Psi}_n\right) - \frac{1}{n} \log|\mathbf{\Psi}_n| - \frac{1}{p} \log|\mathbf{\Theta}_p| + \lambda_1 ||\mathbf{\Psi}_n||_1 + \lambda_2 ||\mathbf{\Theta}_p||_1 \right\}, \tag{6.1}$$

[1] A vector is an order-1 tensor, a matrix is an order-2 tensor and so on.

where $\mathbf{Y}_i$ is the i-th matrix sample, N is the sample size and λ_1, λ_2 the regularization parameters. Minimisation proceeds by fixing one of the precision matrices (say, the columns-precision matrix $\boldsymbol{\Theta}_p$), thus reducing the above to a GLasso problem on $\boldsymbol{\Psi}_n$ with a projected covariance $(\sum_i \mathbf{Y}_i \boldsymbol{\Theta}_p \mathbf{Y}_i^\top)$. Similarly, another GLasso step fits the the columns-precision matrix $\boldsymbol{\Theta}_p$ with a fixed rows-precision $\boldsymbol{\Psi}_n$. Note that each GLasso step involves an additional $\mathcal{O}(N)$ term as the summation depends on a new estimate.

The Kronecker-sum-based BiGLasso Let $\mathbf{Y} \in \mathbb{R}^{n \times p}$ be a random matrix. If its rows are generated as iid samples from $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_p)$, then the sampling distribution of the sufficient statistic $\mathbf{Y}^\top\mathbf{Y}$ is $Wishart(n, \boldsymbol{\Sigma}_p)$ with n degrees of freedom and scale matrix $\boldsymbol{\Sigma}_p$. Similarly, if the columns are generated as iid samples from $\mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}_n)$, then the sampling distribution is $Wishart(p, \boldsymbol{\Gamma}_n)$.

From a *maximum entropy* point of view we can constraint these second-order moments in a model both for the features and the datapoints of a design matrix. One way to do so, is to combine these sufficient statistics in a model for the entire matrix $\mathbf{Y}$ as

$$p(\mathbf{Y}) \;\propto\; \exp\left\{-\mathrm{tr}\left(\boldsymbol{\Psi}_n \mathbf{Y}\mathbf{Y}^\top\right) - \mathrm{tr}\left(\boldsymbol{\Theta}_p \mathbf{Y}^\top \mathbf{Y}\right)\right\} \;,$$

where $\boldsymbol{\Psi}_n \in \mathbb{R}^{n \times n}$ and $\boldsymbol{\Theta}_p \in \mathbb{R}^{p \times p}$ are positive definite matrices. This is equivalent to a joint factorized Gaussian distribution (see eq. (A.6) in the appendix) for the $n \times p$ entries of $\mathbf{Y}$, with a precision matrix of the form

$$\boldsymbol{\Omega} \;\triangleq\; \boldsymbol{\Psi}_n \oplus \boldsymbol{\Theta}_p \;=\; \boldsymbol{\Psi}_n \otimes \mathbf{I}_p + \mathbf{I}_n \otimes \boldsymbol{\Theta}_p \;,$$

where $\otimes$ is the *Kronecker-product* and $\oplus$ the *Kronecker-sum* operator. Thus,

$$\omega_{ij,kl} \;=\; \psi_{i,k}\delta_{j,l} + \delta_{i,k}\theta_{j,l} \;,$$

for $i, k \in \{1, \ldots, n\}$ and $j, l \in \{1, \ldots, p\}$. As an immediate benefit of this parameterization, while the full covariance matrix has $\mathcal{O}(n^2p^2)$ entries, these are governed in our model by only $\mathcal{O}(n^2 + p^2)$ parameters.

Given data in the form of some design matrix $\mathbf{Y}$, the BiGLasso estimates sparse matrices by putting ℓ_1 penalties on $\boldsymbol{\Theta}_p$ and $\boldsymbol{\Psi}_n$. The convex optimization problem is

$$\min_{\boldsymbol{\Theta}_p, \boldsymbol{\Psi}_n} \left\{ n \operatorname{tr}(\boldsymbol{\Theta}_p \mathbf{S}) + p \operatorname{tr}(\boldsymbol{\Psi}_n \mathbf{T}) - \ln|\boldsymbol{\Psi}_n \oplus \boldsymbol{\Theta}_p| + \lambda ||\boldsymbol{\Theta}_p||_1 + \gamma ||\boldsymbol{\Psi}_n||_1 \right\} , \tag{6.2}$$

$$\text{where} \quad \mathbf{S} \triangleq \tfrac{1}{n} \mathbf{Y}^\top \mathbf{Y} \quad \text{and} \quad \mathbf{T} \triangleq \tfrac{1}{p} \mathbf{Y}\mathbf{Y}^\top \tag{6.3}$$

are empirical covariances across the samples and features respectively. A solution simultaneously estimates two graphs – one over the columns of $\mathbf{Y}$, corresponding to the sparsity pattern of $\boldsymbol{\Theta}_p$, and another over the rows of $\mathbf{Y}$, corresponding to the sparsity pattern of $\boldsymbol{\Psi}_n$. Note that (6.2) does not require a summation over the datapoints in each step as was the case in (6.1). Also note that since $\omega_{ii,jj} = \psi_{ii} + \theta_{jj}$, the diagonals of $\boldsymbol{\Theta}_p$ and $\boldsymbol{\Psi}_n$ are not identifiable (though we could restrict the inverses to correlation matrices). However, this does not affect the estimation of the graph *structure* (locations of zeros).

6.3 A penalized likelihood algorithm for BiGLasso

A note on notation If $\mathbf{M}$ is an $np \times np$ matrix written in terms of $p \times p$ blocks, as

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \dots & \mathbf{M}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{M}_{n1} & \dots & \mathbf{M}_{nn} \end{bmatrix} ,$$

then $\operatorname{tr}_p(\mathbf{M})$ is the $n \times n$ matrix of traces of such blocks[1]:

$$\operatorname{tr}_p(\mathbf{M}) = \begin{bmatrix} \operatorname{tr}(\mathbf{M}_{11}) & \dots & \operatorname{tr}(\mathbf{M}_{1n}) \\ \vdots & \ddots & \vdots \\ \operatorname{tr}(\mathbf{M}_{n1}) & \dots & \operatorname{tr}(\mathbf{M}_{nn}) \end{bmatrix} .$$

We alternate between optimizing over $\boldsymbol{\Psi}_n$ while holding $\boldsymbol{\Theta}_p$ fixed and optimiz-

[1] In a sense, this generalizes the conventional trace operator as $\operatorname{tr}_{np}(\mathbf{M}) = \operatorname{tr}(\mathbf{M})$.

ing over $\mathbf{\Theta}_p$ while holding $\mathbf{\Psi}_n$ fixed. First we consider the case where there is no regularization. From (6.2), the first step of the optimization problem is reduced to

$$\min_{\mathbf{\Psi}_n}\left\{p\ \mathrm{tr}\,(\mathbf{\Psi}_n\mathbf{T})\ -\ \ln|\mathbf{\Psi}_n\oplus\mathbf{\Theta}_p|\right\}\ . \tag{6.4}$$

Section A.5 in the supplementary material shows how to take the gradient of (6.4) with respect to $\mathbf{\Psi}_n$. Combining (A.27) and (A.28) of the appendix we obtain the stationary point:

$$\mathbf{T}-\tfrac{1}{2p}\mathbf{T}\circ\mathbf{I}=\tfrac{1}{p}\mathrm{tr}_p(\mathbf{W})-\tfrac{1}{2p}\mathrm{tr}_p(\mathbf{W})\circ\mathbf{I}\ ,$$

where we define $\mathbf{W}\triangleq(\mathbf{\Psi}_n\oplus\mathbf{\Theta}_p)^{-1}$. We partition $\mathbf{V}\triangleq\frac{1}{p}\ \mathrm{tr}_p(\mathbf{W})$ as

$$\mathbf{V}=\begin{bmatrix} v_{11} & \mathbf{v}_{1\backslash 1}^\top \\ \mathbf{v}_{1\backslash 1} & \mathbf{V}_{\backslash 1\backslash 1}\end{bmatrix}\ , \tag{6.5}$$

where $\mathbf{v}_{1\backslash 1}$ is a vector of size $n-1$ and $\mathbf{V}_{\backslash 1\backslash 1}$ is a $(n-1)\times(n-1)$ matrix. Despite the complex form of the stationarity condition, only the lower-left block of its partition will be of use:

$$\begin{aligned}&\mathbf{t}_{1\backslash 1}=\tfrac{1}{p}\mathrm{tr}_p(\mathbf{W}_{1\backslash 1})=\mathbf{v}_{1\backslash 1},\ \text{and also from (6.3)},\\ &\mathbf{t}_{1\backslash 1}=(t_{21},\ldots,t_{n1})^\top=\tfrac{1}{p}(\mathbf{y}_2^\top\mathbf{y}_1,\ \ldots,\mathbf{y}_n^\top\mathbf{y}_1)^\top.\end{aligned} \tag{6.6}$$

Similarly, we partition $\mathbf{W}$ into blocks:

$$\mathbf{W}=\begin{bmatrix} \mathbf{W}_{11} & \mathbf{W}_{1\backslash 1}^\top \\ \mathbf{W}_{1\backslash 1} & \mathbf{W}_{\backslash 1\backslash 1}\end{bmatrix}\ ,$$

where $\mathbf{W}_{11}$ is a $p\times p$ matrix and $\mathbf{W}_{1\backslash 1}$ is a $p(n-1)\times p$ matrix. Then from the bottom-left block of

$$\mathbf{W}\mathbf{\Omega}=\begin{bmatrix} \mathbf{W}_{11} & \mathbf{W}_{1\backslash 1}^\top \\ \mathbf{W}_{1\backslash 1} & \mathbf{W}_{\backslash 1\backslash 1}\end{bmatrix}\begin{bmatrix} \psi_{11}\mathbf{I}_p+\mathbf{\Theta}_p & \ldots & \psi_{in}\mathbf{I}_p \\ \vdots & \ddots & \vdots \\ \psi_{n1}\mathbf{I}_p & \ldots & \psi_{nn}\mathbf{I}_p+\mathbf{\Theta}_p\end{bmatrix}=\mathbf{I}_n\otimes\mathbf{I}_p\ , \tag{6.7}$$

we get

$$\mathbf{W}_{1\backslash 1}(\psi_{11}\mathbf{I}_p + \mathbf{\Theta}_p) + \mathbf{W}_{\backslash 1\backslash 1}(\boldsymbol{\psi}_{1\backslash 1} \otimes \mathbf{I}_p) = \mathbf{0}_{n-1} \otimes \mathbf{I}_p$$

$$\mathbf{W}_{1\backslash 1} + \mathbf{W}_{\backslash 1\backslash 1} \begin{bmatrix} (\psi_{11}\mathbf{I}_p + \mathbf{\Theta}_p)^{-1}\psi_{21} \\ \vdots \\ (\psi_{11}\mathbf{I}_p + \mathbf{\Theta}_p)^{-1}\psi_{n1} \end{bmatrix} = \mathbf{0}_{n-1} \otimes \mathbf{I}_p \tag{6.8}$$

$$\begin{aligned} \mathbf{W}_{1\backslash 1} \ + \ \mathbf{W}_{2\backslash 1}(\psi_{11}\mathbf{I}_p + \mathbf{\Theta}_p)^{-1}\psi_{21} \ + \dots \\ \dots + \ \mathbf{W}_{n\backslash 1}(\psi_{11}\mathbf{I}_p + \mathbf{\Theta}_p)^{-1}\psi_{n1} = \mathbf{0}_{n-1} \otimes \mathbf{I}_p \ , \end{aligned}$$

with $\mathbf{0}_{n-1}$ as the vector of $n-1$ zeros. According to the stationary point in (6.6), taking the blockwise trace $\mathrm{tr}_p(.)$ of both sides, gives the equation:

$$p \ \mathbf{t}_{1\backslash 1} + \mathbf{A}_{\backslash 1\backslash 1}\boldsymbol{\psi}_{1\backslash 1} = \mathbf{0}_{n-1}, \quad \text{where}$$

$$\mathbf{A}_{\backslash 1\backslash 1}^{\top} \triangleq \begin{bmatrix} \mathrm{tr}_p\left\{\mathbf{W}_{2\backslash 1}(\psi_{11}\mathbf{I}_p + \mathbf{\Theta}_p)^{-1}\right\}^{\top} \\ \vdots \\ \mathrm{tr}_p\left\{\mathbf{W}_{n\backslash 1}(\psi_{11}\mathbf{I}_p + \mathbf{\Theta}_p)^{-1}\right\}^{\top} \end{bmatrix} . \tag{6.9}$$

By imposing an ℓ_1 penalty on $\boldsymbol{\psi}_{1\backslash 1}$, this problem reduces to a Lasso regression.

After estimating $\psi_{1\backslash 1}$, we compute $\mathbf{W}_{1\backslash 1}$ by substituting into (6.8). It remains to compute $\mathbf{W}_{11}$. This follows from (6.7), which gives

$$\mathbf{W}_{11} = (\mathbf{I} - \mathbf{W}_{1\backslash 1}^{\top}(\boldsymbol{\psi}_{1\backslash 1} \otimes \mathbf{I}))(\psi_{11}\mathbf{I} + \mathbf{\Theta}_p)^{-1} \ .$$

This algorithm iteratively estimates columns of $\mathbf{\Psi}_n$ and $\mathbf{W}$ in this manner. The procedure for estimating $\mathbf{\Theta}_p$, for fixed $\mathbf{\Psi}_n$, becomes directly parallel to the above simply by *transposing* the design matrix (samples become features and vice-versa) and applying the algorithm. Algorithm 3 outlines the BiGLasso.

In our experiments we treat λ and γ as the same parameter and the precision matrices $\mathbf{\Psi}_n$ and $\mathbf{\Theta}_p$ are initialized as identity matrices. The empirical mean matrix is removed from each dataset.

Algorithm 3 BiGLasso

```
Input: $\mathbf{Y}, \lambda, \gamma$ and initial estimates of $\mathbf{\Psi}_n$ and $\mathbf{\Theta}_p$
$\mathbf{T} \leftarrow p^{-1}\mathbf{Y}\mathbf{Y}^\top$
repeat
    # Estimate $\mathbf{\Psi}_n$ :
    for $i = 1 \dots n$ do
        Partition $\mathbf{\Psi}_n$ into $\psi_{ii}, \boldsymbol{\psi}_{i\setminus i}$ and $\mathbf{\Psi}_{\setminus i \setminus i}$.
        Find a sparse solution of $p\, \mathbf{t}_{i\setminus i} + \mathbf{A}_{\setminus i \setminus i}\boldsymbol{\psi}_{i\setminus i} = \mathbf{0}_{n-1}$ with Lasso regression.
        Substitute $\boldsymbol{\psi}_{i\setminus i}$ into (6.8) to compute $\mathbf{W}_{i\setminus i}$.
        $\mathbf{W}_{ii} \leftarrow \Big(\mathbf{I} - \mathbf{W}_{i\setminus i}^\top(\boldsymbol{\psi}_{i\setminus i} \otimes \mathbf{I})\Big)\,(\psi_{ii}\mathbf{I} + \mathbf{\Theta}_p)^{-1}$
    end for
    # Estimate $\mathbf{\Theta}_p$ :
    Proceed as if estimating $\mathbf{\Psi}_n$ with input $\mathbf{Y}^\top, \lambda, \gamma$.
until (6.2) converges or maximum iterations reached.
```

6.4 Simulations

To empirically assess the efficiency of BiGLasso, we generate the datasets described below from centered Gaussians with Kronecker-product **(KP)** and Kronecker-sum **(KS)** precision matrices. We run the BiGLasso and SMGM using the ℓ_1 penalty. The $\mathbf{\Theta}_p$ and $\mathbf{\Psi}_n$ precision matrices in both cases are generated in accordance to [§4, Leng & Tang, 2012]; namely, as either of the following $d \times d$ blocks (d being either p or n) of increasing density:

1. $\mathbf{A}_1$: Inverse AR(1) (auto-regressive process) such that $\mathbf{A}_1 = \mathbf{B}^{-1}$ with $B_{ij} = 0.7^{|i-j|}$.

2. $\mathbf{A}_2$: AR(4) with $A_{ij} = I(|i-j| = 0) + 0.4I(|i-j| = 1) + 0.2I(|i-j| = 2) + 0.2I(|i-j| = 3) + 0.1I(|i-j| = 4)$, $I(.)$ being the indicator function.

3. $\mathbf{A}_3 = \mathbf{B} + \delta\mathbf{I}$, where for each $B_{ij} = B_{ji}, i \neq j$, $P(B_{ij} = 0.5) = 0.9$ and $P(B_{ij} = 0) = 0.1$. The diagonal is zero and δ is chosen such that the condition number of $\mathbf{A}_3$ is d. Since the condition number is $k(\mathbf{A}_3) = d = \frac{\lambda_1+\delta}{\lambda_d+\delta}$, the ratio of largest-to-smallest eigenvalue, then $\delta = \frac{d\lambda_d-\lambda_1}{1-d}$.

Figures 6.2 and 6.3 show the *recall* $= \frac{\#\{\widehat{\Omega}_{ij}\neq 0 \ \&\ \Omega_{ij}\neq 0\}}{\#\{\Omega_{ij}\neq 0\}}$ (or *true-positive rate*) and *precision* $= \frac{\#\{\widehat{\Omega}_{ij}=0 \ \&\ \Omega_{ij}=0\}}{\#\{\widehat{\Omega}_{ij}=0 \ \&\ \Omega_{ij}=0\}+\#\{\widehat{\Omega}_{ij}=0 \ \&\ \Omega_{ij}=1\}}$ across 50 replications to assess

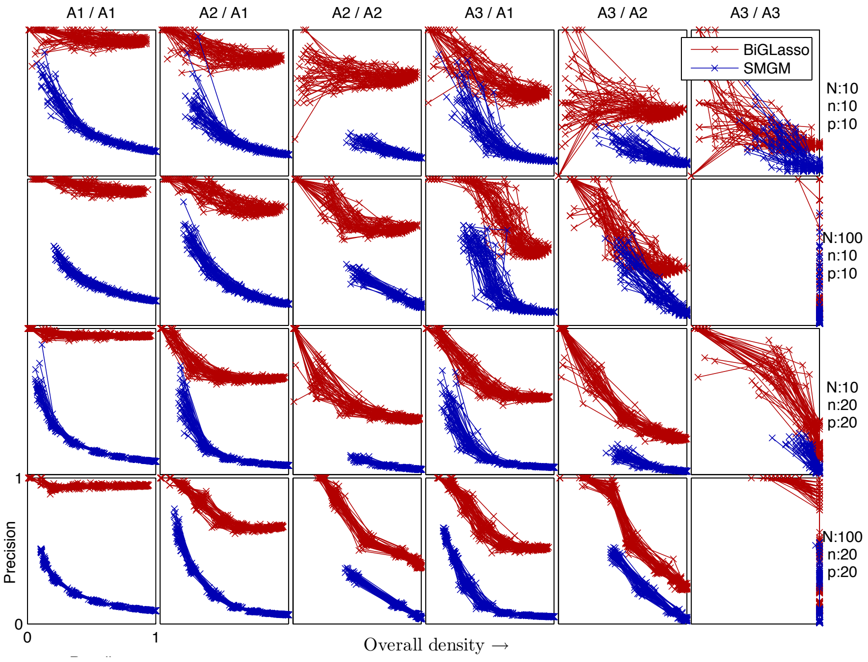


Figure 6.2: Simulation results on data generated from Kronecker-sum structures. Each box shows a recall-precision plot for a particular setup (shown along the top and right margin). Structure recovery can be exact, as the sample size increases for the A3/A3 combination (most right column).

the $\widehat{\mathbf{\Omega}}$ estimates under various setups.

Each box shows a particular setup that varies in block combination $(\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3)$, in block sizes (n, p), in sample size N generated from the matrix-normal and by the structure used (KS or KP) to generate the sample. Each curve in a box is the solution-path of a replication in precision-recall space for a range of regularization settings $\lambda = 5^x$, for $x \in [-6, -2]$ interpolated 10 times. The blocks are arranged such that the overall density of the structured precision matrices increases from left to right.

We note that since blocks A1,A2 have a fixed form, for such combinations each curve is a different sample from the same graph structure. Only A3 is random so in combinations involving A3, each box has a different random A3 and consequently generates a set of 50 replicates from a different graph. At a glance this has little effect.

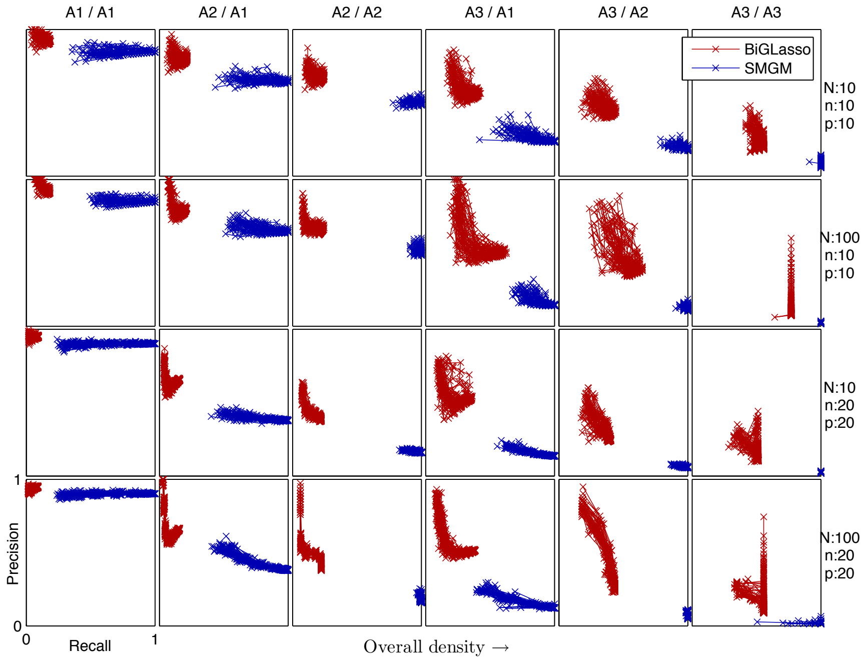


Figure 6.3: Simulation results on data generated from Kronecker-product structures.

Figures 6.2 and 6.3 also compare against the results of SMGM (using the Lasso penalty) on data simulated from the matrix-normal with KS structures. Leng & Tang [2012] had also ran comparisons against the MLE method of Dutilleul [1999] (an unpenalized variant of SMGM), ridge-SMGM (SMGM with an ℓ_2 penalty instead of ℓ_1) and the GLasso of Friedman *et al.* [2008] (on vectorized samples from $\mathcal{N}(\mathbf{0}, \mathbf{\Psi}_n \otimes \mathbf{\Theta}_p)$, i.e. ignoring the matrix structure). They consistently outperformed all of these methods, so for brevity we compare only against the SMGM. Similarly, Figure 6.3 visualizes the simulations under KP structures.

By the empirical distributions of these solution-paths (50 for each model in each box), it is no surprise that the intrinsically denser SMGM tends to have low precision (many false-positives) for smaller values of λ. On the contrary, BiGLasso tends to have low recall (many false-negatives) due to its intrinsically sparser structure.

Block A3 is the only randomized sparse structure whereas A1 and A2 are more "artificial" as they respectively model an inverse-AR(1) and AR(4) and they yield banded precision matrices. Of interest is the observation that the largest effect of the increase in sample size (10 → 100) seems to occur on the A3/A3 combination (right end column of boxes). More precisely in Figure 6.2, we note the difference from box (1,6) to (2,6) and from (3,6) to (4,6). The sample size is very effective: with sufficiently large sample size N, BiGLasso starts to recover exactly and SMGM occupies lower regions in general.

In Figure 6.3, since the data generation process uses Kronecker-product structures, the SMGM is expected to outperform our method. Indeed for lower-density structure, the recovery rate of the SMGM seems consistently better than BiGLasso. and recovery can be almost exact for the SMGM for combination A1/A1. However, as the overall density increases, the performance of BiGLasso is balanced. Again, for combinations involving A3, larger sample sizes benefit BiGLasso more.

In summary, KP-simulated data proved harder to tackle for both methods than KS-generated data. These simulations have shown that the BigLasso consistently outperforms the SMGM on KS-simulations, with the possibility of exact recovery on large sample sizes. On KP-simulations the comparison is less clear, but the BiGLasso proves more practical for denser Kronecker-product structures and the SMGM more practical for sparser structures.

6.5 An example from the COIL dataset

In this section we perform a minor video analysis of a rotating rubber duck from the COIL dataset[1]. The video consists of gray-scaled images, see Figure 6.4. The goal is on two fronts: to recover the conditional dependency structure over the frames and the structure over the pixels. For simplicity, we reduced the resolution of each frame and sub-sampled the frames (at a ratio 1:2). After vectorizing the frames (stacking their columns into 81×1 vectors) and arranging them into a

[1] http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

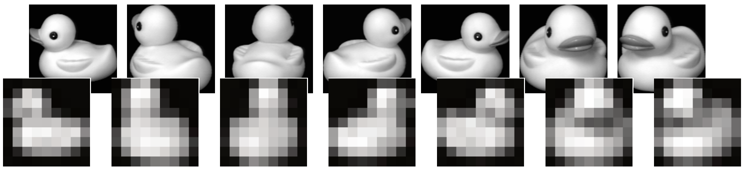


Figure 6.4: Video of a rotating rubber duck. Original resolution of 128 × 128 pixels (back row) and reduced resolution of 9 × 9 pixels (front row).

design matrix $\mathbf{Y}$, the resulting single "datapoint" that BiGLasso has to learn from is 36×81 (#frames × vectorized frame length). Unlike our previous simulations where we had many matrix-samples, here the challenge is to learn from this single matrix ($N = 1$).

Despite the big loss in resolution, the principal component (PCA) subspace of the rotating duck seems to remain smooth, see Figure 6.5. Being a time-series, the video is expected to resemble a 1D manifold, "homeomorphic" to the one recovered by PCA shown in figure 6.5, so we applied the BiGLasso on the reduced images.

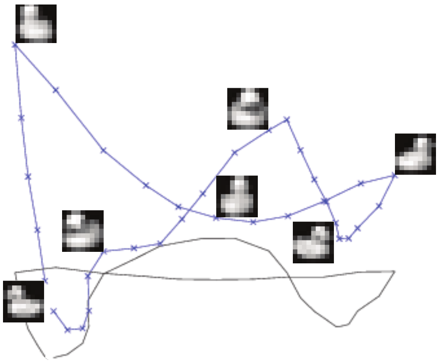


Figure 6.5: 1D manifold of the rotating duck in 3D space, recovered by PCA and projecting onto the 3 principal eigenvectors of $\mathbf{Y}^\top\mathbf{Y}$. The black curve serves as a shadow to aid perspective. Note that the blue line is drawn only by knowledge of the frame ordering and PCA is responsible solely for the reduced embedding.

Indeed, the left panel of figure 6.6 shows the row-precision parameter of BiGLasso capturing a *manifold-like* structure where the first and last frames join, as expected of a 360° rotation. The model recovered the temporal manifold structure, or in other words, we could use it to *connect the dots* in Figure 6.5 in case the true order of the frames was unknown (or randomly given to us).

The right panel of Figure 6.6 shows the conditional dependency structure over the pixels. This figure shows strong dependencies at intervals of 9 — that is, roughly in line with the size of a frame (due to the column-wise ordering of the pixels). This is expected, as neighboring pixels are more likely to be conditionally dependent.

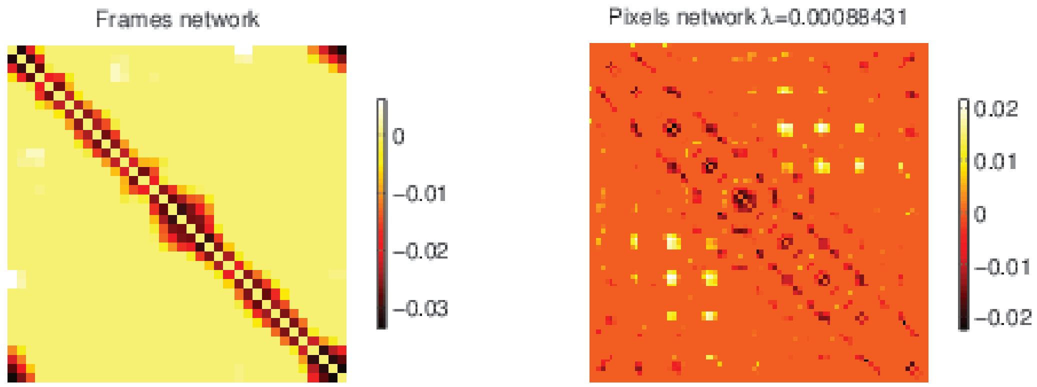


Figure 6.6: Row and column-precision matrix estimates of BiGLasso with $\lambda \approx .0009$.

A more intuitive picture of the induced Markov network is shown in Figure 6.7. A Gaussian graphical model can be naturally interpreted as a system of springs, where the off-diagonal entries of the inverse-covariance represent the *negative stiffness* of the springs. Therefore by the colorbar, a negative-color represents an "attracting" spring between those two pixels and a positive-colour represents a "repulsing" spring. Naturally, in the frames network almost all non-zero elements are negative.

6.6 Summary

There is a need for models to accommodate the growing complexity of dependency structures. We are concerned with *conditional* dependencies, as encoded

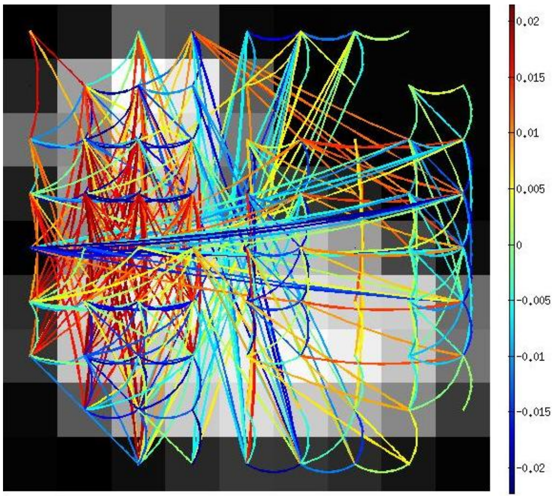


Figure 6.7: The Markov network induced by the column-precision over the pixels (superimposed over the first frame for reference of the pixel locations).

by the inverse-covariance of a matrix-normal density. In high-dimensional cases the Markov network structures induced by a graph could be approximated by factorisations such as the tensor-product (Kronecker-product of precision matrices). In this work, we motivated a novel application of the Cartesian factorization of graphs (Kronecker-sum of precision matrices), as a more parsimonious and interpretable structure for inter-sample and inter-feature conditional dependencies. In the context of multi-output GPs, the Kronecker-product cancels any transfer (that is, ignoring any correlations) between outputs (tasks) when a block design with a noise-free covariance. This is not the case with the Kronecker-sum due to its additive form. We introduced the bigraphical Lasso, an algorithm for the simultaneous point-estimation of the structures of two Gaussian graphical models: one over the rows of a matrix-sample and the other over its columns. This was demonstrated to good effect through simulations as well as a toy example from the COIL dataset.

An obvious extension that would exploit the associativity of the Cartesian product, would be the modeling of datasets organised into 3 or higher-dimensional arrays (amounting to GRFs over higher-order tensors) with dependencies across any subset of the array dimensions.

One of the appealing features of the Kronecker-sum of precision matrices is the *preservation* of inter-task transfer, thereby leading to potential applications on Kronecker-sums of kernels for multi-output Gaussian processes.

Finally we feel that the — largely unknown to machine learning — literature on the Cartesian product of graphs deserves a thorough study, towards modeling and algorithmic advances in probabilistic graphical modeling.

Authors contributions

The chapter is based on a manuscript version written by Alfredo Kalaitzis, along with the literature review and motivations. Experiments were designed and experimental results written by AK. John Lafferty and Neil Lawrence devised the model and JL wrote the initial algorithm for optimising the penalised likelihood.

Chapter 7

Conclusions and future work

Conclusions

In the thesis we have studied a number of covariance structures for exploiting relationships between covariates. The relevancy of these structure to realistic problems relies on three properties of the Gaussian distribution: the non-parametric formulation of smooth functions with kernel matrices, the formulation of conditional independence constraints through structural zeros in the inverse-covariance and most importantly, the formulation of low-rank bases through the spectral analysis of the covariance.

In chapter 2 we proposed the Gaussian process as the default tool for fitting gene-expression trajectories with encouraging results compared to a state-of-the-art Bayesian hierarchical model specialised to this task. The inter-timepoint modeling was done through a temporal covariance structure, defined by a RBF for simplicity. Other kernels could help, but the question of preference would be lie outside the scope of this thesis. The residual structure left by the RBF was assumed to be Gaussian spherical noise.

The relaxation of this assumption motivated chapter 3, where we introduced the *residual component analysis* (RCA) algorithm: a maximum-likelihood approach for identifying a low dimensional representation of the residuals when the covariance is partially explained by another covariance of fixed-effects. We proved how the low-rank component in the joint covariance of the covariates can be determined through a generalized eigenvalue problem (GEP), for dual and primal representations. With further analysis, the GEP of RCA turned out to reduce to PCA/PPCA (a regular eigenvalue problem) on the joint sample-covariance of a particular linear transformation of the data. We also showed that this transformation is strongly connected to an oblique (non-orthogonal) projection of the data. The projector is governed by the inverse of the explained covariance term, which plays the role of a *null-steering operator* in the posterior expectation of the latent components. Conversely for a fixed transformation, every PCA problem is mapped to an RCA problem, effectively declaring them equivalent. Now every problem that can be reduced to RCA, can also be reduced to PCA.

Chapter 4 enumerated a few such problems, with the most prominent recently in machine learning being CCA/PCCA. Another notable example is LDA, which was already known to reduce to CCA (so that connection came for free), though it involves mixed (binary and continuous) data. The primal and dual variants of the RCA theorem provide a probabilistic interpretation to classical generalised-projection algorithms, thus with the potential to unify many different algorithms. The take-away message was that with further imaginative instantiations of the explained covariance term $\mathbf{\Sigma}$, one can develop new approaches to data analysis.

A few such approaches are demonstrated in Chapter 5. One approach involves a more accurate fitting of sparse-inverses (or structure learning of Gaussian graphical models) by combining it with a low-rank covariance term that acts as the residual covariance when the data suffer from confounding low-rank effects. With the former problem having already been solved to some extent by the Graphical Lasso and the latter with a just-proposed solution, the natural next step was to devise an EM algorithm (EM/RCA) that iteratively fits one structure at a time. The EM/RCA algorithm was tested to good effect on protein-signaling data, motion-capture data and Eurovision voting data. The second new approach revisited the GP regression problem from Chapter 2 with a low-rank plus RBF kernel covariance structure to characterise more accurately any the structured error in the time-series. Again, the idea was to fit both structures in a flip-flop fashion, but a single pass sufficed to outperform the same baseline method from chapter 2.

Finally in chapter 6, we focused to *conditional* independencies induced by the inverse-covariance of a matrix-normal density. The vectorised re-parametrization of the matrix-normal has a Kronecker-product covariance structure. Its properties enable flip-flop approaches for fitting its $np \times np$ covariance that reduce training time from $\mathcal{O}(n^2p^2)$ to $\mathcal{O}(n^2 + p^2)$ in the number of matrix dimensions. We motivated a novel structure for the joint precision, the Cartesian factorisation of graphs (Kronecker-sum of precisions), as a more parsimonious and interpretable structure for inter-sample and inter-feature conditional dependencies. In the context of regression, the Kronecker-product cancels any transfer (that is, inducing zero correlations) between responses (tasks) of multiple regressions. The Kronecker-sum does not suffer from this shortcoming due to its additive functional form. We also proposed the bi-Graphical Lasso (BiGLasso), an algorithm with the novel Kronecker-sum inverse-covariance structure for the simultaneous L1-estimation of the structures of two Gaussian graphical models: one over the rows of a matrix-sample and the other over its columns. It responded with encouraging results on simulations as well as an example from the COIL dataset.

Future work

We touched on a number of potential directions while discussing RCA.

Better inference One could enrich the RCA framework in ways parallel to the developments of PPCA (for instance, see Bayesian PCA [Bishop, 1999]). A Bayesian treatment of RCA would be of interest as there is a plethora of work on priors for the low-rank part that consider the latent dimensionality of the confounders, or the sparsity of the low-rank weights $\mathbf{W}$, for instance see *spike-and-slab* [Mohamed *et al.*, 2012], the *horseshoe* [Carvalho *et al.*, 2010], or the *generalised double Pareto* prior [Armagan *et al.*, 2011]). Chapter 5 presented a MAP estimator for the M-step (of EM/RCA) when the explained covariance term $\mathbf{\Sigma}$ is sparse-inverse, but one might wish to consider suitable priors depending on the application (and thus structure) of $\mathbf{\Sigma}$. These suggestions would merely augment our earlier graphical models.

Other structured composites Also mentioned in §3.2.3, a combination of *low-rank plus sparse* might be useful for learning *marginal dependency* structures, for a Bayesian approach see [Silva, 2011]. The Kronecker-sum structure studied in Chapter 6 can be readily extended to three or more precisions by exploiting the associativity of the Cartesian product. Training on 3-or-higher-dimensional data-arrays (amounting to the number of indices required to access an entry) would proceed analogously to Algorithm 3 but with more transpositions of the data-array indices involved.

Non-linear non-Gaussian models Finally, we established a solid relationship of RCA[1] to PCA and CCA, but not so much to LDA, ICA and their *kernelised* variants. It is not straightforward how one could generalise RCA to a GPLVM-like model, as oblique projections probably lose any meaning in an RKHS in general. However, we believe there is hope for representing mixed-data (discrete, ordinal, continuous) via Gaussian copulas — an up-and-coming line of research for semi-parametric learning in machine learning and more traditional in statistics. Existing approaches could be adapted [Hoff, 2007; Murray *et al.*, 2011] so that our RCA framework is applied on the Gaussian latent representations of non-Gaussian, discrete or ordinal observations.

[1] Usage of the term 'RCA' here is akin to PCA/PPCA and refers collectively to the RCA generalised eigenvalue problem and RCA maximum-likelihood solution.

Appendix A

Mathematical background

In this appendix we provide enough basic background on each topic to make this thesis self-contained. For further discussions on each topic we provide references in their respective sections.

A.1 Gaussian identities

Let $X = \{x_1, x_2, ..., x_n\}$ be a set of scalar random variables in $\mathbb{R}$,

$$x_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right) \ .$$

If $\{A, B\}$ is a partition on X, that is, $A \cup B = X$ and $A \cap B = \emptyset$, for non-empty A and B, then with $\mathbf{x}_A$ we denote an ordered collection of the random variables in A. By slight abuse of notation, we use $\mathbf{x}$ also as a vector of scalar random variables in $\mathbb{R}^p$. Thereby,

$$\mathcal{N}\left(\mathbf{x}_A | \ \boldsymbol{\mu}_A, \boldsymbol{\Sigma}_A\right) = (2\pi)^{-p/2} |\boldsymbol{\Sigma}_A|^{-1/2} \exp\left\{-\tfrac{1}{2}(\mathbf{x}_A - \boldsymbol{\mu}_A)^\top \boldsymbol{\Sigma}_A^{-1} (\mathbf{x}_A - \boldsymbol{\mu}_A)\right\} \ ,$$

denotes the Gaussian distribution over the random variable $\mathbf{x}_A$ and similarly for the set B. The *joint*, *marginal* and *conditional* distributions of Gaussians are also Gaussian but the *product* of two Gaussians distribution yields an *unnormalised* Gaussian. See also [Bishop, 2006; Von Mises, 1964] for a detailed treatment.

A.1.1 Joint distribution

If $\mathbf{x}_A \sim \mathcal{N}(\boldsymbol{\mu}_A, \boldsymbol{\Sigma}_A)$ and $\mathbf{x}_B \sim \mathcal{N}(\boldsymbol{\mu}_B, \boldsymbol{\Sigma}_B)$, then

$$\begin{bmatrix} \mathbf{x}_A \\ \mathbf{x}_B \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_A & \boldsymbol{\Sigma}_{AB} \\ \boldsymbol{\Sigma}_{BA} & \boldsymbol{\Sigma}_B \end{bmatrix} \right) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Lambda}_A & \boldsymbol{\Lambda}_{AB} \\ \boldsymbol{\Lambda}_{BA} & \boldsymbol{\Lambda}_B \end{bmatrix}^{-1} \right) , \tag{A.1}$$

where the *partitioned* joint precision (inverse of the joint covariance) is: $\begin{bmatrix} \boldsymbol{\Lambda}_A & \boldsymbol{\Lambda}_{AB} \\ \boldsymbol{\Lambda}_{BA} & \boldsymbol{\Lambda}_B \end{bmatrix}$

$$= \begin{bmatrix} \left(\boldsymbol{\Sigma}_A - \boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_B^{-1}\boldsymbol{\Sigma}_{BA}\right)^{-1} & -\boldsymbol{\Lambda}_A\boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_B^{-1} \\ -\boldsymbol{\Sigma}_B^{-1}\boldsymbol{\Sigma}_{BA}\boldsymbol{\Lambda}_A & \boldsymbol{\Sigma}_B^{-1} + \boldsymbol{\Sigma}_B^{-1}\boldsymbol{\Sigma}_{BA}\boldsymbol{\Lambda}_A\boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_B^{-1} \end{bmatrix} . \tag{A.2}$$

A.1.2 Marginal distribution

If eq. (A.1) holds for two random variables $\mathbf{x}_A$ and $\mathbf{x}_b$, then the marginal distribution of $\mathbf{x}_A$ is:

$$p(\mathbf{x}_A) = \int \mathrm{d}\mathbf{x}_B \; p(\mathbf{x}_A, \mathbf{x}_B) = \mathcal{N}(\boldsymbol{\mu}_A, \boldsymbol{\Sigma}_A) \tag{A.3}$$

and similarly for $\mathbf{x}_B$. Marginalisation relies on completing a square in the exponential of the Gaussian, see [Bishop, 2006].

A.1.3 Conditional distribution

More interestingly, the mean and covariance of the conditional distribution of $\mathbf{x}_A|\mathbf{x}_B$ are:

$$\boldsymbol{\mu}_{A|B} = \boldsymbol{\mu}_A + \boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_B^{-1}(\mathbf{x}_B - \boldsymbol{\mu}_B) \tag{A.4}$$

$$\boldsymbol{\Sigma}_{A|B} = \boldsymbol{\Lambda}_A^{-1} \ . \tag{A.5}$$

Note that the *conditional precision* is simply the upper-left block of the *joint precision*, eq. (A.2). This nicely juxtaposes the *marginal covariance* being simply the upper-left block of the *joint covariance*, eq. (A.3).

A.1.4 Product of Gaussians

The product of two Gaussian distributions over the same domain yields an *unnormalised* Gaussian:

$$\begin{aligned} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_A, \boldsymbol{\Sigma}_A)\, \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_B, \boldsymbol{\Sigma}_B) \ \propto \ \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_C, \boldsymbol{\Sigma}_C)\,, &\quad \text{where} \\ \boldsymbol{\mu}_C = \boldsymbol{\Sigma}_C \left(\boldsymbol{\Sigma}_A^{-1}\boldsymbol{\mu}_A + \boldsymbol{\Sigma}_B^{-1}\boldsymbol{\mu}_B\right)^{-1} & \\ \boldsymbol{\Sigma}_C = \left(\boldsymbol{\Sigma}_A^{-1} + \boldsymbol{\Sigma}_B^{-1}\right)^{-1} \ . & \end{aligned} \tag{A.6}$$

For the product, note that the precision matrix of the unnormalised Gaussian is simply the *sum* of the individual precisions and the mean is the *convex sum* of the means, weighted by the individual precisions [Rasmussen & Williams, 2006, section A.2].

A.2 Matrix derivatives

All matrix derivatives are based on the following differential forms ($\mathbf{a}, \mathbf{A}$ are constants):

$$\begin{aligned}
\partial\mathbf{A} &= \mathbf{0} && \text{(A.7)}\\
\partial(\mathbf{X}^\top) &= (\partial\mathbf{X})^\top && \text{(A.8)}\\
\partial(\mathbf{X}+\mathbf{Y}) &= \partial\mathbf{X}+\partial\mathbf{Y} && \text{(A.9)}\\
\partial(\mathbf{A}\mathbf{X}) &= \mathbf{A}\partial\mathbf{X} && \text{(A.10)}\\
\partial(\mathbf{a}^\top\mathbf{X}\mathbf{a}) &= \mathbf{a}^\top(\partial\mathbf{X})\mathbf{a} && \text{(A.11)}\\
\partial(\mathbf{X}\mathbf{Y}) &= (\partial\mathbf{X})\mathbf{Y}+\mathbf{X}(\partial\mathbf{Y}) && \text{(A.12)}\\
\partial(\mathbf{X}\otimes\mathbf{Y}) &= (\partial\mathbf{X})\otimes\mathbf{Y}+\mathbf{X}\otimes(\partial\mathbf{Y}) && \text{(A.13)}\\
\partial\mathbf{X}^{-1} &= -\mathbf{X}^{-1}(\partial\mathbf{X})\mathbf{X}^{-1} && \text{(A.14)}\\
\partial\,\mathrm{tr}\,(\mathbf{X}) &= \mathrm{tr}\,(\partial\mathbf{X}) && \text{(A.15)}\\
\partial\ln|\mathbf{X}| &= \mathrm{tr}\left(\mathbf{X}^{-1}\partial\mathbf{X}\right)\ . && \text{(A.16)}
\end{aligned}$$

For proofs on these identities, see [Magnus & Neudecker, 1988]. For any of the above, if the $\mathbf{X}$ in $\frac{\partial f}{\partial\mathbf{X}}$ is symmetric then

$$\frac{\partial f}{\partial\mathbf{X}} = \left[\frac{\partial f}{\partial\mathbf{X}}\right] + \left[\frac{\partial f}{\partial\mathbf{X}}\right]^\top - \mathbf{I}\circ\left[\frac{\partial f}{\partial\mathbf{X}}\right]\ . \tag{A.17}$$

A.3 Linear algebra

We describe here some basic matrix properties for reference [Golub & Van Loan, 1996; Horn & Johnson, 1990; Strang, 2003]. In the following, let $\mathbf{X} \in \mathbb{R}^{n\times m}$ be a arbitrary real rectangular matrix.

A.3.1 Singular value decomposition

As a culmination of the *fundamental theorem of linear algebra*, the SVD neatly connects the *four fundamental subspaces* of $\mathbf{X}$ in the form of mapping between two orthonormal spaces:

$$\mathbf{X} = \mathbf{U}\mathbf{L}\mathbf{V}^\top \ , \tag{A.18}$$

where $\mathbf{U}$ is an orthogonal matrix whose columns are called the *right-singular vectors* and they form an orthonormal basis for the *direct sum*[1] *of the column-space and left-null space* of $\mathbf{X}$. The columns of the orthogonal matrix $\mathbf{V}$ are the *left-singular vectors* and they form a basis for the *direct sum of the row-space and null space*. The diagonal $m \times n$ matrix $\mathbf{L}$ contains the *singular values* l_i responsible for the scaling from one space to the other. Now, any linear transformation can be broken down to its constituent steps: For a right-singular vector $\mathbf{v}_i$,

$$\mathbf{X}\mathbf{v}_i = (\mathbf{U}\mathbf{L}\mathbf{V}^\top)\mathbf{v}_i = \mathbf{U}\mathbf{L}\mathbf{e}_i = l_i\mathbf{u}_i \ ,$$

and since any $\mathbf{a} \in \mathbb{R}^m$ can be expressed as a linear combination of the right-singular vectors $\mathbf{v}_i$, then

$$\mathbf{X}\mathbf{a} = (\mathbf{U}\mathbf{L}\mathbf{V}^\top)(c_1\mathbf{v}_1 + \cdots + c_m\mathbf{v}_m) = \sum_i c_i l_i \mathbf{u}_i = \mathbf{b} \ ,$$

shows the mapping to a linear combination of the left-singular vectors $\mathbf{u}_i$.

Connection to the spectral theorem

It follows that the left and right-singular vectors are the *eigenvectors* of $\mathbf{X}\mathbf{X}^\top$ and $\mathbf{X}^\top\mathbf{X}$ respectively and the singular values are the *square roots of the eigenvalues*,

[1] $\mathcal{X} \oplus \mathcal{Y} = \{(x, y) \mid x \in \mathcal{X}, y \in \mathcal{Y}\}$.

shared by both matrices:

$$\begin{aligned}\mathbf{X}\mathbf{X}^\top &= (\mathbf{U}\mathbf{L}\mathbf{V}^\top)(\mathbf{V}\mathbf{L}^\top\mathbf{U}^\top) = \mathbf{U}\mathbf{L}^2\mathbf{U}^\top \\ \mathbf{X}^\top\mathbf{X} &= (\mathbf{V}\mathbf{L}^\top\mathbf{U}^\top)(\mathbf{U}\mathbf{L}\mathbf{V}^\top) = \mathbf{V}\mathbf{L}^2\mathbf{V}^\top \ .\end{aligned}$$

If $\mathbf{X}$ is symmetric then obviously $\mathbf{X}\mathbf{X}^\top = \mathbf{X}^\top\mathbf{X} = \mathbf{X}^2 = \mathbf{U}\mathbf{L}^2\mathbf{U}^\top = \mathbf{V}\mathbf{L}^2\mathbf{V}^\top$, thereby $\mathbf{U} = \mathbf{V}$ and

$$\mathbf{X} = \mathbf{U}\mathbf{L}\mathbf{U}^\top \ , \tag{A.19}$$

which is the *spectral* or *eigen-decomposition* of $\mathbf{X}$. This also shows a method to compute the singular vectors of any $\mathbf{X}$ through the eigenvectors and eigenvalues of $\mathbf{X}\mathbf{X}^\top$ and $\mathbf{X}^\top\mathbf{X}$, which in turn can be computed efficiently by the *QR method* or *Francis QR step* [Golub & Van Loan, 1996]. In the special case where all the eigenvalues of the symmetric $\mathbf{X}$ are positive (non-negative), then $\mathbf{X}$ is *positive (semi-)definite*, that is, $\mathbf{a}^\top\mathbf{X}\mathbf{a} > \ 0$ $(\geq \ 0)$ for all non-zero $\mathbf{a} \in \mathbb{R}^m$. Therefore, $\mathbf{X}\mathbf{X}^\top$ and $\mathbf{X}^\top\mathbf{X}$ are *always* positive (at-least-)semi-definite.

Economy SVD

For matrices $\mathbf{X} \in \mathbb{R}^{n\times m}$ of rank r where m and n differ greatly, an equivalent *economic* form of the SVD should be used:

$$\mathbf{X} = \mathbf{U}_r\mathbf{L}_r\mathbf{V}_r^\top = \sum_i l_i\,\mathbf{u}_i\,\mathbf{v}_i^\top \ ,$$

where $\mathbf{U}_r$ and $\mathbf{V}_r$ now have $r \leq \min(n, m)$ orthonormal columns, thus we only have to compute the r principal eigenvalues and eigenvectors.

A.3.2 Geometry of oblique projections

The notation and introduction approach in this section are adapted from [Behrens & Scharf, 1994]. Let $\mathbf{P} \in \mathbb{R}^{p\times p}$ be an arbitrary projection matrix with column rank $q < p$. Let $\mathcal{C}(\mathbf{P})$, $\mathcal{R}(\mathbf{P})$ and $\mathcal{N}(\mathbf{P})$ denote the *column-space* (or *range*), *row-space* and *null-space* of $\mathbf{P}$ respectively. What we mean by a *projection* is that the matrix transforms any vector $\mathbf{x}$ in $\mathbb{R}^p$ to a vector inside the column-space $\mathcal{C}(\mathbf{P})$ of dimension q. Naturally, any vector already in $\mathcal{C}(\mathbf{P})$ is unaffected by $\mathbf{P}$, therefore for $\mathbf{P}$ to be a projection it must be *idempotent*:

$$\mathbf{P}^2 = \mathbf{P} \ .$$

An eigenvector of the projection is either any vector in $\mathcal{C}(\mathbf{P})$, with corresponding eigenvalue 1, or any vector in the space that is orthogonal to the row space $\mathcal{R}(\mathbf{P})$, that is, the null space $\mathcal{N}(\mathbf{P})$, with eigenvalue 0. Now we make a distinction between *orthogonal* and *oblique* (non-orthogonal) projections. For this, let $\mathbf{X} \in \mathbb{R}^{p \times q}$ be an arbitrary rectangular matrix of rank q.

Orthogonal projections Intuitively, *orthogonal* projections are characterised by the fact that they project the whole of $\mathbb{R}^p$ on a *right* vertically to $\mathcal{C}(\mathbf{P})$. Hence, only vectors of the space *orthogonal* to the column space $\mathcal{C}(\mathbf{P})$, the null space $\mathcal{N}(\mathbf{P})$, are mapped to $\mathbf{0}$. The projection $\mathbf{P}$ is orthogonal iff

$$\mathbf{P} = \mathbf{P}^\top \ .$$

To construct an orthogonal projection that projects onto the column-space of $\mathbf{X}$ ($\mathcal{C}(\mathbf{P}) = \mathcal{C}(\mathbf{X})$), then

$$\mathbf{P}_\mathbf{X} \triangleq \mathbf{X} \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \ ,$$

which is symmetric and idempotent. Note that $\mathbf{P}_\mathbf{X}\mathbf{X} = \mathbf{X}$ and $\mathbf{P}_\mathbf{X}\mathbf{A} = \mathbf{0}$, where $\mathcal{C}(\mathbf{A}) = \mathcal{N}(\mathbf{X})$ and that only a basis of $\mathcal{C}(\mathbf{X})$ is needed to uniquely define $\mathbf{P}_\mathbf{X}$.

Oblique projections The projection angle of $\mathbf{P}$ wrt to $\mathcal{C}(\mathbf{P})$ is *oblique* (non-right) iff $\mathbf{P}$ is *not symmetric*, thought it is still idempotent. Recall that for an orthogonal projection, since the angle projecting on $\mathcal{C}(\mathbf{X})$ is right, then the column space of the projection *automatically determines the space along which* $\mathbf{P}$ projects. In other words, there is only one possible choice for $\mathcal{N}(\mathbf{P})$ and that is $\mathcal{N}(\mathbf{X})$. *This is not the case for oblique projections.* Due to the projecting direction forming an oblique angle α[1] with $\mathcal{C}(\mathbf{P})$, there is an infinity of possible choices for $\mathcal{N}(\mathbf{P})$: simply take any such space with the angle α from $\mathcal{C}(\mathbf{X})$ and precess it about the axis orthogonal to $\mathcal{C}(\mathbf{X})$. Thereby, we need *two spaces* to uniquely define an oblique projection $\mathbf{P}_{\mathbf{X},\mathbf{N}}$, $\mathcal{C}(\mathbf{P}) \triangleq \mathcal{C}(\mathbf{X})$ and $\mathcal{N}(\mathbf{P}) \triangleq \mathcal{C}(\mathbf{N})$, for some $\mathbf{N} \in \mathbb{R}^{p \times q'}$, such that $q' + q \leq p$, whose columns we arrange to span the null-space of the oblique projection. Now note that $\mathbf{P}_{\mathbf{X},\mathbf{N}}\mathbf{X} = \mathbf{X}$ and $\mathbf{P}_{\mathbf{X},\mathbf{N}}\mathbf{N} = \mathbf{0}$. In this sense, oblique projections can be seen to *generalise*[2] orthogonal projections as they depend on a superset of the parameters. Figure A.1 illustrates the relationship between the fundamental subspaces of projections.

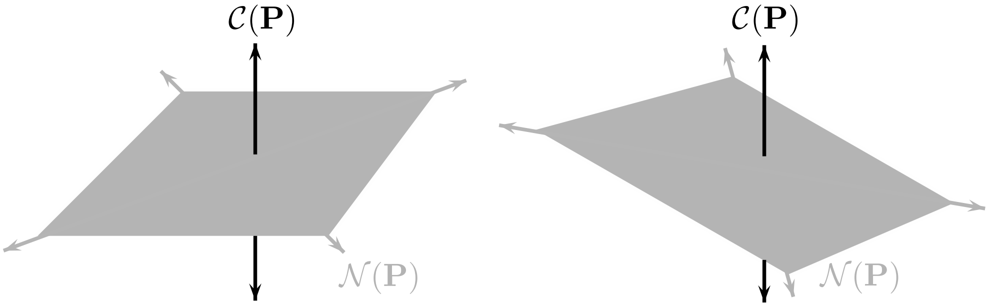

$\mathcal{C}(\mathbf{P})$
$\mathcal{N}(\mathbf{P})$
$\mathcal{C}(\mathbf{P})$
$\mathcal{N}(\mathbf{P})$

Figure A.1: The fundamental subspaces of projections. The column-space and null-space of an orthogonal projection (left panel) are *orthogonal complements* so either one uniquely determines the other. The null-space of an oblique projection (right panel) intersects the column-space at the origin at an *oblique* angle, so both bases are required to characterise the projection onto $\mathcal{C}(\mathbf{P})$ along the direction of $\mathcal{N}(\mathbf{P})$. Note that in this case $\mathcal{N}(\mathbf{P})$ can have any dimension $q' \leq p$; in the diagram the subspaces jointly span the whole space purely for visualisation purposes.

Now consider the joint subspace spanned by the concatenated columns of

[1] Actually, there is a *set* of principal angles between two Euclidean spaces, but for simplicity we refer to them as one.

[2] Not in the strict sense since they do not subsume orthogonal projections.

$[\mathbf{X}\ \mathbf{N}]$, hence the partitioned form of its orthogonal projection is

$$\mathbf{P}_{[\mathbf{X}\ \mathbf{N}]} \triangleq [\mathbf{X}\ \mathbf{N}] \begin{bmatrix} \mathbf{X}^\top\mathbf{X} & \mathbf{X}^\top\mathbf{N} \\ \mathbf{N}^\top\mathbf{X} & \mathbf{N}^\top\mathbf{N} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X}^\top \\ \mathbf{N}^\top \end{bmatrix} .$$

In the formula, we can see that the left block provides a basis that is coupled with coordinates computed by the middle and right blocks. Then we can decompose the contributions of $\mathcal{C}(\mathbf{X})$ and $\mathcal{C}(\mathbf{N})$ since they are disjoint by assumption:

$$\begin{aligned} \mathbf{P}_{[\mathbf{X}\ \mathbf{N}]} &= [\mathbf{X}\ \mathbf{0}] \begin{bmatrix} \mathbf{X}^\top\mathbf{X} & \mathbf{X}^\top\mathbf{N} \\ \mathbf{N}^\top\mathbf{X} & \mathbf{N}^\top\mathbf{N} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X}^\top \\ \mathbf{N}^\top \end{bmatrix} + [\mathbf{0}\ \mathbf{N}] \begin{bmatrix} \mathbf{X}^\top\mathbf{X} & \mathbf{X}^\top\mathbf{N} \\ \mathbf{N}^\top\mathbf{X} & \mathbf{N}^\top\mathbf{N} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X}^\top \\ \mathbf{N}^\top \end{bmatrix} \\ &= \mathbf{P}_{\mathbf{X},\mathbf{N}} + \mathbf{P}_{\mathbf{N},\mathbf{X}} . \end{aligned}$$

Using the *partitioned inverse formula* gives the more concise form:

$$\mathbf{P}_{\mathbf{X},\mathbf{N}} = \mathbf{X}(\mathbf{X}^\top\mathbf{P}_{\mathbf{N}}^\perp\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{P}_{\mathbf{N}}^\perp , \tag{A.20}$$

where $\mathbf{P}_{\mathbf{N}}$ is the projector onto $\mathcal{C}(\mathbf{N})$ and $\mathbf{P}_{\mathbf{N}}^\perp \triangleq \mathbf{I} - \mathbf{P}_{\mathbf{N}}$ the projector onto the orthogonal complement of $\mathcal{C}(\mathbf{N})$. The projector $\mathbf{P}_{\mathbf{N}}^\perp$ can be seen as a null-steering operator that nulls everything in the null-space of the projector. It is easy to verify that $\mathbf{P}$ is *idempotent* (but not symmetric) with column-space $\mathcal{C}(\mathbf{X})$: $\mathbf{P}_{\mathbf{X},\mathbf{N}}\mathbf{X} = \mathbf{X}(\mathbf{X}^\top\mathbf{P}_{\mathbf{N}}^\perp\mathbf{X})^{-1}(\mathbf{X}^\top\mathbf{P}_{\mathbf{N}}^\perp\mathbf{X}) = \mathbf{X}$, and null space $\mathcal{C}(\mathbf{N})$: $\mathbf{P}_{\mathbf{X},\mathbf{N}}\mathbf{N} = \mathbf{X}(\mathbf{X}^\top\mathbf{P}_{\mathbf{N}}^\perp\mathbf{X})^{-1}\mathbf{X}^\top(\mathbf{P}_{\mathbf{N}}^\perp\mathbf{N}) = \mathbf{0}$. It follows that $\mathbf{P}_{\mathbf{X},\mathbf{N}}$ is the oblique projection onto $\mathcal{C}(\mathbf{X})$ along the direction of $\mathcal{C}(\mathbf{N})$ and similarly for $\mathbf{P}_{\mathbf{N},\mathbf{X}}$.

A.3.3 Woodbury matrix identity

In a partitioned inverse, such as eq. (A.2), the inverse of the upper-left block $\mathbf{\Lambda}_A^{-1} = \mathbf{\Sigma}_A - \mathbf{\Sigma}_{AB}\mathbf{\Sigma}_B^{-1}\mathbf{\Sigma}_{BA}$, is known as the *Schur complement* of $\mathbf{\Sigma}_B$ in the joint covariance matrix. An alternative way to partition the inverse is through the Schur complement of $\mathbf{\Sigma}_A$, that is, $\mathbf{\Lambda}_B^{-1} = \mathbf{\Sigma}_B - \mathbf{\Sigma}_{BA}\mathbf{\Sigma}_A^{-1}\mathbf{\Sigma}_{AB}$, (which requires only an exchange of the letters):

$$\boldsymbol{\Sigma}^{-1} = \begin{bmatrix} \boldsymbol{\Sigma}_A^{-1} + \boldsymbol{\Sigma}_A^{-1}\boldsymbol{\Sigma}_{AB}\boldsymbol{\Lambda}_B\boldsymbol{\Sigma}_{BA}\boldsymbol{\Sigma}_A^{-1} & -\boldsymbol{\Lambda}_B\boldsymbol{\Sigma}_{BA}\boldsymbol{\Sigma}_A^{-1} \\ -\boldsymbol{\Sigma}_A^{-1}\boldsymbol{\Sigma}_{AB}\boldsymbol{\Lambda}_B & \left(\boldsymbol{\Sigma}_B - \boldsymbol{\Sigma}_{BA}\boldsymbol{\Sigma}_A^{-1}\boldsymbol{\Sigma}_{AB}\right)^{-1} \end{bmatrix} . \tag{A.21}$$

Equating the upper-left blocks of eqs. (A.2) and (A.21) gives the *Woodbury matrix identity* (in its general form for symmetric matrices):

$$\left(\mathbf{A} - \mathbf{C}\mathbf{B}^{-1}\mathbf{C}^\top\right)^{-1} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{C}\left(\mathbf{B} - \mathbf{C}^\top\mathbf{A}^{-1}\mathbf{C}\right)^{-1}\mathbf{C}^\top\mathbf{A}^{-1} , \tag{A.22}$$

where $\mathbf{A} = \boldsymbol{\Sigma}_A, \quad \mathbf{B} = \boldsymbol{\Sigma}_B \quad \text{and} \quad \mathbf{C} = \boldsymbol{\Sigma}_{AB}$.

A.4 EM for learning low-rank plus sparse-inverse covariance structures

A.4.1 Variational lower bound

Given data $\mathbf{Y}$, our goal is to infer the sparse structure of the underlying GMRF, encoded by a sparse-inverse covariance term $\boldsymbol{\Lambda}^{-1}$. We can efficiently estimate $\boldsymbol{\Lambda}$ with the GLASSO algorithm [Banerjee *et al.*, 2008; Friedman *et al.*, 2008]. The challenge is to estimate $\boldsymbol{\Lambda}$ *in the presence of low-rank structures* $\mathbf{W}\mathbf{W}^\top$ in the marginal covariance, induced by confounders $\mathbf{X}$. In a fully Bayesian setting we would compute the posterior $p(\boldsymbol{\Lambda} \,|\, \mathbf{Y}, \mathbf{W})$. By Bayes' rule:

$$p(\boldsymbol{\Lambda} \,|\, \mathbf{Y}, \mathbf{W}) \propto p(\mathbf{Y} \,|\, \boldsymbol{\Lambda}, \mathbf{W}) \; p(\boldsymbol{\Lambda}) \; ,$$

where $p(\boldsymbol{\Lambda})$ is some kind of sparsity-inducing prior on $\boldsymbol{\Lambda}$ (for instance, a Laplace distribution). The normalising constant of the posterior is an intractable integral so it is omitted. We opt for a MAP (point) estimate of $\boldsymbol{\Lambda}$ which is equivalent to maximising the joint distribution $p(\mathbf{Y}, \boldsymbol{\Lambda} \,|\, \mathbf{W}) = p(\mathbf{Y} \,|\, \boldsymbol{\Lambda}, \mathbf{W}) \; p(\boldsymbol{\Lambda})$. However, the maximum has no closed-form solution so we approximate it by *maximising a lower bound* on the mode of the log-posterior:

$$\begin{aligned} \ln \left\{ p(\mathbf{Y}, \boldsymbol{\Lambda} \,|\, \mathbf{W}) \right\} &= \ln \int p(\mathbf{Y}, \boldsymbol{\Lambda}, \mathbf{Z} \,|\, \mathbf{W}) \; \mathrm{d}\mathbf{Z} = \ln \int q(\mathbf{Z}) \; \frac{p(\mathbf{Y}, \boldsymbol{\Lambda}, \mathbf{Z} \,|\, \mathbf{W})}{q(\mathbf{Z})} \; \mathrm{d}\mathbf{Z} \\ &\geq \int q(\mathbf{Z}) \; \ln \left\{ \frac{p(\mathbf{Y}, \boldsymbol{\Lambda}, \mathbf{Z} \,|\, \mathbf{W})}{q(\mathbf{Z})} \right\} \; \mathrm{d}\mathbf{Z} \; . \end{aligned} \tag{A.23}$$

Since the log function is *concave*, the last line applies *Jensen's inequality* to yield the lower bound [MacKay, 2003].

A.4.2 Update equations

Now we derive the update equations of the hybrid EM/RCA algorithm for optimising the parameters of the low-rank plus sparse-inverse covariance in the joint distribution $p(\mathbf{Y}, \boldsymbol{\Lambda} \,|\, \mathbf{W}) = \prod_i \mathcal{N}\left(\mathbf{y}_i \,|\, \mathbf{0}, \mathbf{W}\mathbf{W}^\top + \boldsymbol{\Lambda}^{-1}\right) \; p(\boldsymbol{\Lambda})$. For fixed $\boldsymbol{\Lambda}'$ and $\mathbf{W}'$ the lower bound is maximised when the variational distribution equals the posterior: $q(\mathbf{Z}) = p(\mathbf{Z} \,|\, \mathbf{Y}, \mathbf{W}', \boldsymbol{\Lambda}')$. We have:

$$\begin{aligned} &\ln \left\{ p(\mathbf{Z} \,|\, \mathbf{Y}, \mathbf{W}', \boldsymbol{\Lambda}') \right\} \\ &\overset{c}{=} \ln p(\mathbf{Y} \,|\, \mathbf{Z}, \mathbf{W}') + \ln p(\mathbf{Z} \,|\, \boldsymbol{\Lambda}') = \ln \mathcal{N}\left(\mathbf{Y} \,|\, \mathbf{Z}, \mathbf{W}'\mathbf{W}'^\top + \sigma^2 \mathbf{I}\right) + \ln \mathcal{N}\left(\mathbf{Z} \,|\, \mathbf{0}, \boldsymbol{\Lambda}'^{-1}\right) \\ &\overset{c}{=} \tfrac{1}{2} \sum_i \left\{ -\ln |\mathbf{W}'\mathbf{W}'^\top + \sigma^2 \mathbf{I}| |\boldsymbol{\Lambda}'| - (\mathbf{y}_i - \mathbf{z}_i)^\top \left(\mathbf{W}'\mathbf{W}'^\top + \sigma^2 \mathbf{I}\right)^{-1} (\mathbf{y}_i - \mathbf{z}_i) - \left(\mathbf{z}_i^\top \boldsymbol{\Lambda}' \mathbf{z}_i\right) \right\} . \end{aligned}$$

Update for Z Isolating the *linear* and *quadratic* terms in $\mathbf{z}_i$ gives the posterior expectation and covariance respectively as the update equations for $\mathbf{Z}$:

$$\operatorname{var}[\mathbf{z} \,|\, \mathbf{y}] = \left(\left(\mathbf{W}'\mathbf{W}'^{\top} + \sigma^2 \mathbf{I} \right)^{-1} + \boldsymbol{\Lambda}' \right)^{-1} \tag{A.24}$$

$$\mathbb{E}[\mathbf{z}_i \,|\, \mathbf{y}_i] = \operatorname{var}[\mathbf{z} \,|\, \mathbf{y}] \left(\mathbf{W}'\mathbf{W}'^{\top} + \sigma^2 \mathbf{I} \right)^{-1} \mathbf{y}_i \tag{A.25}$$

$$\mathbb{E}_{p(\mathbf{z} \,|\, \mathbf{y})}[\mathbf{z}_i \mathbf{z}_i^{\top}] = \operatorname{var}[\mathbf{z} \,|\, \mathbf{y}] \; + \; \mathbb{E}[\mathbf{z}_i \,|\, \mathbf{y}_i] \, \mathbb{E}[\mathbf{z}_i \,|\, \mathbf{y}_i]^{\top} \; . \tag{A.26}$$

By fixing the variational distribution as the new posterior $q(\mathbf{Z}) = p(\mathbf{Z} \,|\, \mathbf{Y}, \mathbf{W}', \boldsymbol{\Lambda}')$, eq. (A.23) becomes

$$\begin{aligned} & \int p(\mathbf{Z} \,|\, \mathbf{Y}, \mathbf{W}', \boldsymbol{\Lambda}') \; \ln \left\{ \frac{p(\mathbf{Z} \,|\, \mathbf{Y}, \mathbf{W}', \boldsymbol{\Lambda}') \; p(\mathbf{Y}, \boldsymbol{\Lambda}' \,|\, \mathbf{W}')}{p(\mathbf{Z} \,|\, \mathbf{Y}, \mathbf{W}', \boldsymbol{\Lambda}')} \right\} \; \mathrm{d}\mathbf{Z} \\ = & \int p(\mathbf{Z} \,|\, \mathbf{Y}, \mathbf{W}', \boldsymbol{\Lambda}') \; \ln p(\mathbf{Y}, \boldsymbol{\Lambda}' \,|\, \mathbf{W}') \; \mathrm{d}\mathbf{Z} \quad = \quad \ln p(\mathbf{Y}, \boldsymbol{\Lambda}' \,|\, \mathbf{W}') \; , \end{aligned}$$

the maximisation of which wrt $\boldsymbol{\Lambda}$ leads to its update (the M-step).

Update for Λ From eq. (A.23), isolating any factors that depend on $\boldsymbol{\Lambda}$ gives:

$$\begin{aligned} & \int p(\mathbf{Z} \,|\, \mathbf{Y}, \mathbf{W}', \boldsymbol{\Lambda}') \; \ln \left\{ \frac{p(\mathbf{Z}, \boldsymbol{\Lambda}) \; p(\mathbf{Y} \,|\, \mathbf{Z}, \mathbf{W}')}{p(\mathbf{Z} \,|\, \mathbf{Y}, \mathbf{W}', \boldsymbol{\Lambda}')} \right\} \; \mathrm{d}\mathbf{Z} \\ = & \int p(\mathbf{Z} \,|\, \mathbf{Y}, \mathbf{W}', \boldsymbol{\Lambda}') \; \{ \ln p(\mathbf{Z}, \boldsymbol{\Lambda}) + \ln p(\mathbf{Y} \,|\, \mathbf{Z}, \mathbf{W}') - \ln p(\mathbf{Z} \,|\, \mathbf{Y}, \mathbf{W}', \boldsymbol{\Lambda}') \} \; \mathrm{d}\mathbf{Z} \; . \end{aligned}$$

The maximisation of the above does not depend on terms independent from $\mathbf{\Lambda}$. Eliminating such terms leads to the complete-data expected log-likelihood:

$$\begin{aligned}
&\underset{\mathbf{\Lambda}}{\operatorname{argmax}} \int p(\mathbf{Z}\,|\,\mathbf{Y},\mathbf{W}',\mathbf{\Lambda}')\ \ln p(\mathbf{Z},\mathbf{\Lambda})\ \mathrm{d}\mathbf{Z} \quad = \quad \underset{\mathbf{\Lambda}}{\operatorname{argmax}}\ \mathbb{E}_{p(\mathbf{Z}\,|\,\mathbf{Y})}\left[\ln p(\mathbf{Z},\mathbf{\Lambda})\right] \\
= \quad & \underset{\mathbf{\Lambda}}{\operatorname{argmax}}\ \mathbb{E}_{p(\mathbf{Z}\,|\,\mathbf{Y})}\left[\sum_i \left\{-\tfrac{p}{2}\ln 2\pi + \tfrac{1}{2}\ln|\mathbf{\Lambda}| - \tfrac{1}{2}\mathbf{z}_i^\top \mathbf{\Lambda}\mathbf{z}_i\right\} - \tfrac{n}{2}\lambda||\mathbf{\Lambda}||_1\right] \\
= \quad & \underset{\mathbf{\Lambda}}{\operatorname{argmax}}\ \ln|\mathbf{\Lambda}| - \tfrac{1}{n}\sum_i \left\{\mathbb{E}_{p(\mathbf{Z}\,|\,\mathbf{Y})}\left[\operatorname{tr}\left(\mathbf{z}_i\mathbf{z}_i^\top\mathbf{\Lambda}\right)\right]\right\} - \lambda||\mathbf{\Lambda}||_1 \\
= \quad & \underset{\mathbf{\Lambda}}{\operatorname{argmax}}\ \ln|\mathbf{\Lambda}| - \operatorname{tr}\left(\tfrac{1}{n}\sum_i \left\{\mathbb{E}_{p(\mathbf{Z}\,|\,\mathbf{Y})}\left[\mathbf{z}_i\mathbf{z}_i^\top\right]\right\}\mathbf{\Lambda}\right) - \lambda||\mathbf{\Lambda}||_1\ .
\end{aligned}$$

By eq. (A.26), the last line amounts to a GLASSO problem with covariance

$$\begin{aligned}
\tfrac{1}{n}\sum_i \mathbb{E}_{p(\mathbf{Z}\,|\,\mathbf{Y})}\left[\mathbf{z}_i\mathbf{z}_i^\top\right] &= \operatorname{var}\left[\mathbf{z}\,|\,\mathbf{y}\right] \ + \ \tfrac{1}{n}\widehat{\mathbf{Z}}^\top\widehat{\mathbf{Z}}\ , \\
\text{where} \quad \widehat{\mathbf{Z}}^\top &= \left[\ \mathbb{E}\left[\mathbf{z}_1\,|\,\mathbf{y}_1\right]\ \ldots\ \mathbb{E}\left[\mathbf{z}_n\,|\,\mathbf{y}_n\right]\ \right]\ .
\end{aligned}$$

A.5 Derivatives for BiGLasso

Gradient wrt $\boldsymbol{\Psi}_n$ Taking the gradient of eq. (6.4), with respect to $\boldsymbol{\Psi}_{ij}$ and using the identity (A.16) we get:

$$\begin{aligned}\frac{\partial}{\partial \Psi_{ij}} \ln |\boldsymbol{\Psi}_n \oplus \boldsymbol{\Theta}_p| &= \mathrm{tr}\left\{ (\boldsymbol{\Psi}_n \oplus \boldsymbol{\Theta}_p)^{-1} \frac{\partial (\boldsymbol{\Psi}_n \oplus \boldsymbol{\Theta}_p)}{\partial \Psi_{ij}} \right\} \\ &= \mathrm{tr}\left\{ \mathbf{W} \left(\frac{\partial \boldsymbol{\Psi}_n}{\partial \Psi_{ij}} \otimes \mathbf{I}_p \right) \right\}, \quad \text{by} \quad (\text{A.13}) \\ &= \mathrm{tr}\left\{ \mathbf{W} \left((\mathbf{J}^{ij} + \mathbf{J}^{ji} - \mathbf{J}^{ij}\mathbf{J}^{ij}) \otimes \mathbf{I}_p \right) \right\}, \quad \text{by} \quad (\text{A.17})\end{aligned}$$

$$= \mathrm{tr}\left\{ \mathbf{W} \begin{bmatrix} \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \mathbf{I}_p^{(\mathbf{i},\mathbf{j})} & \vdots \\ \mathbf{0} & \dots & \mathbf{0} \end{bmatrix} \right\} + \mathrm{tr}\left\{ \mathbf{W} \left(\mathbf{J}^{ji} \otimes \mathbf{I}_p \right) \right\} - \mathrm{tr}\left\{ \mathbf{W} \left(\mathbf{J}^{ij}\mathbf{J}^{ij} \otimes \mathbf{I}_p \right) \right\}$$

$$= 2 \ \mathrm{tr}\left\{ \mathbf{W}_{(\mathbf{i},\mathbf{j})} \right\} - \delta_{ij} \mathrm{tr}\left\{ \mathbf{W}_{(\mathbf{i},\mathbf{j})} \right\},$$

where $\mathbf{W} \triangleq (\boldsymbol{\Psi}_n \oplus \boldsymbol{\Theta}_p)^{-1}$; $\mathbf{I}_p^{(\mathbf{i},\mathbf{j})}$ is at the (i,j)-th block of size $p \times p$, that is, $(\mathbf{i},\mathbf{j}) = [(pi - p + 1) : pi\,, \ (pj - p + 1) : pj]$; $\mathbf{J}^{ij}$ is the single-entry matrix (with $J_{ij} = 1$ and zeros elsewhere); $\delta_{ij} = 1$ if $i = j$, $\delta_{ij} = 0$ if $i \neq j$. Therefore

$$\frac{\partial}{\partial \boldsymbol{\Psi}_n} \ln |\boldsymbol{\Psi}_n \oplus \boldsymbol{\Theta}_p| = 2 \ \mathrm{tr}_p(\mathbf{W}) - \mathrm{tr}_p(\mathbf{W}) \circ \mathbf{I}\,. \tag{A.27}$$

Also, using (A.10) and (A.15) gives

$$\frac{\partial \ p \, \mathrm{tr}(\boldsymbol{\Psi}_n \mathbf{T})}{\partial \boldsymbol{\Psi}_n} = 2p \, \mathbf{T} - \mathbf{T} \circ \mathbf{I}\,. \tag{A.28}$$

References

Agakov, F., Orchard, P. & Storkey, A. (2012). Discriminative mixtures of sparse latent fields for risk management. In *Proceedings of AISTATS 2012*. 85

Agarwal, A. & Triggs, B. (2006). Recovering 3D human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**. 76

Akaho, S. (2001). A kernel method for canonical correlation analysis. In *In Proceedings of the International Meeting of the Psychometric Society (IMPS2001)*. 72

Angelini, C., De Canditiis, D., Mutarelli, M. & Pensky, M. (2007). A Bayesian approach to estimation and testing in time-course microarray experiments. *Stat Appl Genet Mol Biol*, **6**, 24. 7, 8, 22, 27, 96

Angelini, C., Cutillo, L., De Canditiis, D., Mutarelli, M. & Pensky, M. (2008). BATS: a Bayesian user-friendly software for analyzing time series microarray experiments. *BMC bioinformatics*, **9**, 415. 7, 8, 27

Armagan, A., Dunson, D. & Lee, J. (2011). Generalized double Pareto shrinkage. *arXiv preprint arXiv:1104.0861*. 120

Azar, Y., Fiat, A., Karlin, A., McSherry, F. & Saia, J. (2001). Spectral analysis of data. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, 619–626, ACM. 71

Bach, F.R. & Jordan, M.I. (2002). Kernel independent component analysis. *Journal of Machine Learning Research*, **3**, 1–48. 49, 60, 66, 70, 72, 73

Bach, F.R. & Jordan, M.I. (2005). A probabilistic interpretation of canonical correlation analysis. Tech. Rep. 688, Department of Statistics, University of California, Berkeley. 49

Banerjee, O., El Ghaoui, L. & d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *The Journal of Machine Learning Research*, **9**, 485–516. 4, 48, 82, 84, 100, 101, 131

Bansal, M., Gatta, G.D. & Di Bernardo, D. (2006). Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. *Bioinformatics*, **22**, 815. 7

Bar-Joseph, Z. (2004). Analyzing time series gene expression data. *Bioinformatics*, **20**, 2493. 7

Bar-Joseph, Z., Gerber, G., Simon, I., Gifford, D.K. & Jaakkola, T.S. (2003). Comparing the continuous representation of time-series expression profiles to identify differentially expressed genes. *Proceedings of the National Academy of Sciences of the United States of America*, **100**, 10146. 7

Bartholomew, D., Knott, M. & Moustaki, I. (2011). *Latent variable models and factor analysis: a unified approach*. Wiley. 40, 69

Basilevsky, A.T. (1994). *Statistical factor analysis and related methods: theory and applications*. Wiley Series in Probability and Statistics, Wiley, Hoboken, NJ. 40, 69

Bay, S.D., Chrisman, L., Pohorille, A. & Shrager, J. (2004). Temporal aggregation bias and inference of causal regulatory networks. *Journal of Computational Biology*, **11**, 971–985. 8

Behrens, R. & Scharf, L. (1994). Signal processing applications of oblique projection operators. *Signal Processing, IEEE Transactions on*, **42**, 1413–1424. 51, 61, 127

Berlinet, A. & Thomas-Agnan, C. (2004). *Reproducing kernel Hilbert spaces in probability and statistics*. Springer. 17

Binkley, J. & Nelson, C. (1988). A note on the efficiency of seemingly unrelated regression. *The American Statistician*, **42**, 137–139. 104

Bishop, C.M. (1999). Bayesian PCA. In M.J. Kearns, S.A. Solla & D.A. Cohn, eds., *Advances in Neural Information Processing Systems*, vol. 11, 482–388, MIT Press, Cambridge, MA. 60, 120

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag. 58, 123

Bonilla, E.V., Chai, K.M.A. & Williams, C.K.I. (2008). Multi-task Gaussian process prediction. In J.C. Platt, D. Koller, Y. Singer & S. Roweis, eds., *Advances in Neural Information Processing Systems*, vol. 20, MIT Press, Cambridge, MA. 103

Browne, M. (1979). The maximum-likelihood solution in inter-battery factor analysis. *British Journal of Mathematical and Statistical Psychology*, **32**, 75–86. 50, 67, 74

Carvalho, C., Polson, N. & Scott, J. (2010). The horseshoe estimator for sparse signals. *Biometrika*, **97**, 465–480. 120

Chandrasekaran, V., Parrilo, P.A. & Willsky, A.S. (2010). Latent variable graphical model selection via convex optimization. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, 1610–1613, IEEE. 85

Chen, M. & Wang, Z. (2006). Subspace tracking in colored noise based on oblique projection. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 3, III–III, IEEE. 50, 61

Chung, F.R.K. (1996). *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society. 103

Dalal, N. & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, 886 –893 vol. 1. 77

Damianou, A., Titsias, M.K. & Lawrence, N.D. (2011). Variational Gaussian process dynamical systems. In P. Bartlett, F. Peirrera, C. Williams & J. Lafferty, eds., *Advances in Neural Information Processing Systems*, vol. 24, MIT Press, Cambridge, MA. 44

Dawid, A.P. (1981). Some matrix-variate distribution theory: notational considerations and a Bayesian application. *Biometrika*, **68**, 265–274. 101

De Bie, T., Cristianini, N. & Rosipal, R. (2005). Eigenproblems in pattern recognition. *Handbook of Geometric Computing: Applications in Pattern Recognition, Computer Vision, Neuralcomputing, and Robotics*, 129–170. 48, 65, 70

Della Gatta, G., Bansal, M., Ambesi-Impiombato, A., Antonini, D., Missero, C. & di Bernardo, D. (2008). Direct targets of the TRP63 transcription factor revealed by a combination of gene expression profiling and reverse engineering. *Genome research*, **18**, 939. 7, 8, 26, 30, 31, 95, 96

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**, 1–38. 84

Duda, R.O. & Hart, P.E. (1973). *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York. 69

Dudoit, S., Yang, Y.H., Callow, M.J. & Speed, T.P. (2002). Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Statistica sinica*, **12**, 111–140. 6

Dutilleul, P. (1999). The mle algorithm for the matrix normal distribution. *Journal of statistical computation and simulation*, **64**, 105–123. 101, 112

Efron, B., Tibshirani, R., Storey, J.D. & Tusher, V. (2001). Empirical Bayes analysis of a microarray experiment. *Journal of the American Statistical Association*, **96**, 1151–1160. 6

Ek, C.H., Rihan, J., Torr, P., Rogez, G. & Lawrence, N.D. (2008). Ambiguity modeling in latent spaces. In A. Popescu-Belis & R. Stiefelhagen, eds., *Machine Learning for Multimodal Interaction (MLMI 2008)*, LNCS, 62–73, Springer-Verlag. 45, 50, 75

Ernst, J., Nau, G. & Bar-Joseph, Z. (2005). Clustering short time series gene expression data. *Bioinformatics*, **21**, i159. 7

Fan, J. & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, **96**, 1348–1360. 101

Finkenstadt, B., Heron, E.A., Komorowski, M., Edwards, K., Tang, S., Harper, C.V., Davis, J.R.E., White, M.R.H., Millar, A.J. & Rand, D.A. (2008). Reconstruction of transcriptional dynamics from gene reporter data using differential equations. *Bioinformatics*, **24**, 2901. 7

Fisher, R.A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, **7**, 179–188, reprinted in *Contributions to Mathematical Statistics*, John Wiley: New York (1950). 70

Friedman, J., Hastie, T. & Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, **9**, 432–441. 4, 48, 82, 84, 101, 112, 131

Friedman, N., Linial, M., Nachman, I. & Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of computational biology*, **7**, 601–620. 6

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, CA, 2nd edn. 69

Fusi, N., Stegle, O. & Lawrence, N.D. (2012). Joint modelling of confounding factors and prominent genetic regulators provides increased accuracy in genetical genomics studies. *PLoS Computat Biol*, **8**, e1002330. 45, 47

Gao, P., Honkela, A., Rattray, M. & Lawrence, N.D. (2008). Gaussian process modelling of latent chemical species: applications to inferring transcription factor activities. *Bioinformatics*, **24**, i70. 10

Golub, G. & Van Loan, C. (1996). *Matrix computations*, vol. 3. Johns Hopkins Univ Pr. 125, 127

Gupta, A.K. & Nagar, D.K. (1999). *Matrix variate distributions*. Chapman Hill. 101

Hastie, T., Tibshirani, R. & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer-Verlag, 2nd edn. 49, 58

Hoff, P. (2007). Extending the rank likelihood for semiparametric copula estimation. *The Annals of Applied Statistics*, 265–283. 121

Honkela, A., Girardot, C., Gustafson, E.H., Liu, Y.H., Furlong, E.E.M., Lawrence, N.D. & Rattray, M. (2010). Model-based method for transcription factor target identification with limited data. *Proceedings of the National Academy of Sciences*, **107**, 7793. 7

Horn, R. & Johnson, C. (1990). *Matrix analysis*. Cambridge University Press. 65, 125

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, **24**, 417–441. 40, 65

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, **28**, 321–377. 40, 65

Hu, Y. & Loizou, P. (2003). A generalized subspace approach for enhancing speech corrupted by colored noise. *Speech and Audio Processing, IEEE Transactions on*, **11**, 334 – 341. 50, 61

Imrich, W., Klavzar, S. & Rall, D.F. (2008). *Topics in Graph Theory: Graphs and Their Cartesian Product*. AK Peters Ltd. 103

Irizarry, R.A., Hobbs, B., Collin, F., Beazer-Barclay, Y.D., Antonellis, K.J., Scherf, U. & Speed, T.P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, **4**, 249. 30

Jolliffe, I.T. (2002). *Principal component analysis*, vol. 2. Wiley Online Library. 40, 58

Kalaitzis, A.A. & Lawrence, N.D. (2011a). Residual component analysis. Tech. rep., University of Sheffield, arXiv report. 4, 36

Kalaitzis, A.A. & Lawrence, N.D. (2011b). A simple approach to ranking differentially expressed gene expression time courses through Gaussian process regression. *BMC Bioinformatics*, **12**. 2, 5, 97

Kalaitzis, A.A. & Lawrence, N.D. (2012). Residual component analysis. In J. Langford & J. Pineau, eds., *Proceedings of the International Conference in Machine Learning*, vol. 29, Morgan Kauffman, San Francisco, CA. 4, 36

Kannan, R., Vempala, S. & Vetta, A. (2004). On clusterings: Good, bad and spectral. *Journal of the ACM*, **51**, 497–515. 71

Kerr, M.K., Martin, M. & Churchill, G.A. (2000). Analysis of variance for gene expression microarray data. *Journal of Computational Biology*, **7**, 819–837. 6

Kettenring, J.R. (1971). Canonical analysis of several sets of variables. *Biometrika*, **58**, 433–451. 73

Kirk, P.D.W. & Stumpf, M.P.H. (2009). Gaussian process regression bootstrapping: exploring the effects of uncertainty in time course data. *Bioinformatics*, **25**, 1300. 10

Klami, A. & Kaski, S. (2006). Generative models that discover dependencies between data sets. In *Machine Learning for Signal Processing, 2006. Proceedings of the 2006 16th IEEE Signal Processing Society Workshop on*, 123 –128. 3, 45, 50, 67, 74

Klami, A. & Kaski, S. (2007). Local dependent components. In *Proceedings of the 24th international conference on Machine learning*, 425–432, ACM. 60, 74

Klami, A. & Kaski, S. (2008). Probabilistic approach to detecting dependencies between data sets. *Neurocomputing*, **72**, 39–46. 50, 67, 74

Lai, P.L. & Fyfe, C. (2001). Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, **10**, 365–377. 72

Lauritzen, S.L. (1996). *Graphical models*, vol. 17. Oxford University Press, USA. 47, 82, 101

Lawrence, N.D. (2004). Gaussian process latent variable models for visualisation of high dimensional data. In S. Thrun, L. Saul & B. Schölkopf, eds., *Advances in Neural Information Processing Systems 16*, 329, MIT Press, Cambridge, MA. 44

Lawrence, N.D. (2005). Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, **6**, 1783–1816. 42, 44, 60

Lawrence, N.D. (2012). A unifying probabilistic perspective for spectral dimensionality reduction: Insights and new models. *Journal of Machine Learning Research*, **13**. 100

Lawrence, N.D., Sanguinetti, G. & Rattray, M. (2007). Modelling transcriptional regulation using Gaussian processes. *Advances in Neural Information Processing Systems*, **19**, 785. 10

Lawrence, N.D., Girolami, M., Rattray, M. & Sanguinetti, G., eds. (2010). *Learning and Inference in Computational Systems Biology*. MIT Press, Cambridge, MA. 84

Leng, C. & Tang, C. (2012). Sparse matrix graphical models. *Journal of the American Statistical Association*, **107**, 1187–1200. 101, 105, 110, 112

Lönnstedt, I. & Speed, T.P. (2002). Replicated microarray data. *Statistica Sinica*, **12**, 31–46. 6

MacKay, D. (1998). Introduction to Gaussian processes. *NATO ASI Series F Computer and Systems Sciences*, **168**, 133–166. 16

MacKay, D.J.C. (1999). Comparison of approximate methods for handling hyperparameters. *Neural Computation*, **11**, 1035–1068. 22

MacKay, D.J.C. (2003). *Information theory, inference, and learning algorithms*. Cambridge University Press. 9, 11, 13, 34, 132

Magnus, J. & Neudecker, H. (1988). *Matrix differential calculus with applications in statistics and econometrics*. Wiley. 125

Meinshausen, N. & Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **72**, 417–473. 86

Mohamed, S., Heller, K. & Ghahramani, Z. (2012). Bayesian and L1 approaches for sparse unsupervised learning. In J. Langford & J. Pineau, eds., *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, ICML '12, 751–758, Omnipress, New York, NY, USA. 120

Möller, M.F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks*, **6**, 525–533. 21

Murray, J., Dunson, D., Carin, L. & Lucas, J. (2011). Bayesian Gaussian copula factor models for mixed data. *arXiv preprint arXiv:1111.0317*. 121

Neal, R.M. (1997). Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. Tech. rep., arXiv report. 22

Ng, A.Y., Jordan, M.I. & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In T.G. Dietterich, S. Becker & Z. Ghahramani, eds., *Advances in Neural Information Processing Systems*, vol. 14, MIT Press, Cambridge, MA. 71

O'Hagan, A. (1998). A markov property for covariance structures. *Statistics Research Report 98-13, Nottingham University*. 104

Parlett, B.N. (1980). *The Symmetric Eigenvalue Problem*. Prentice-Hall, Englewood Cliffs, N.J. 65

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, **2**, 559–572. 40

Pinheiro, J. & Bates, D. (2000). *Mixed-effects models in S and S-PLUS*. Springer Verlag. 45, 80

Rasmussen, C.E. & Williams, C.K.I. (2006). *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA. 2, 9, 11, 13, 15, 17, 25, 43, 124

Richardson, T. & Spirtes, P. (2002). Ancestral graph Markov models. *The Annals of Statistics*, **30**, 962–1030. 49

Roweis, S. & Ghahramani, Z. (1999). A unifying review of linear Gaussian models. *Neural computation*, **11**, 305–345. 1

Roweis, S.T. (1998). EM algorithms for PCA and SPCA. In M.I. Jordan, M.J. Kearns & S.A. Solla, eds., *Advances in Neural Information Processing Systems*, vol. 10, 626–632, MIT Press, Cambridge, MA. 38

Roweis, S.T. & Saul, L.K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, **290**, 2323–2326. 100

Rubin, D.B. & Thayer, D.T. (1982). EM algorithms for ML factor analysis. *Psychometrika*, **47**, 69–76. 69

Sabidussi, G. (1959). Graph multiplication. *Mathematische Zeitschrift*, **72**, 446–457, 10.1007/BF01162967. 103

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D.A. & Nolan, G.P. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, **308**, 523–529. 87, 88

Schena, M., Shalon, D., Davis, R.W. & Brown, P.O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, **270**, 467. 8

Schölkopf, B. & Smola, A. (2002). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. the MIT Press. 17

Schölkopf, B., Smola, A.J. & Müller, K.R. (1997). Kernel principal component analysis. In *Proceedings 1997 International Conference on Artificial Neural Networks, ICANN'97*, 583, Lausanne, Switzerland. 60

SCHWARZ, G. (1978). Estimating the dimension of a model. *The annals of statistics*, **6**, 461–464. 77

SILVA, R. (2011). A MCMC approach for Learning the Structure of Gaussian Acyclic Directed Mixed Graphs. In *Proceedings of the 8th Scientific Meeting of the Classification and Data Analysis Group of the Italian Statistical Society (invited paper), CLADAG 2011*. 49, 121

SPELLMAN, P.T., SHERLOCK, G., ZHANG, M.Q., IYER, V.R., ANDERS, K., EISEN, M.B., BROWN, P.O., BOTSTEIN, D. & FUTCHER, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular biology of the cell*, **9**, 3273. 6

STEGLE, O., DENBY, K.J., WILD, L., MCHATTIE, S., MEADE, A., GHAHRAMANI, Z. & BORGWARDT, K.M. (2009). Discovering temporal patterns of differential gene expression in microarray time series. In *GCB*, 133–142. 10

STEGLE, O., DENBY, K.J., COOKE, E.J., WILD, D.L., GHAHRAMANI, Z. & BORGWARDT, K.M. (2010). A robust Bayesian two-sample test for detecting intervals of differential gene expression in microarray time series. *Journal of Computational Biology*, **17**, 355–367. 8, 10, 22, 33, 34

STEGLE, O., LIPPERT, C., MOOIJ, J., LAWRENCE, N. & BORGWARDT, K. (2011). Efficient inference in matrix-variate Gaussian models with iid observation noise. *Advances in Neural Information Processing Systems*, **24**, 443. 49, 87, 104

STOREY, J.D., XIAO, W., LEEK, J.T., TOMPKINS, R.G. & DAVIS, R.W. (2005). Significance analysis of time course microarray experiments. *Proceedings of the National Academy of Sciences of the United States of America*, **102**, 12837. 7

STRANG, G. (2003). *Introduction to linear algebra*. Wellesley Cambridge Pr. 125

SUBRAMANIAN, A., TAMAYO, P., MOOTHA, V.K., MUKHERJEE, S., EBERT, B.L., GILLETTE, M.A., PAULOVICH, A., POMEROY, S.L., GOLUB, T.R., LANDER, E.S. *et al.* (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, **102**, 15545. 30

SUN, L., JI, S. & YE, J. (2011). Canonical correlation analysis for multilabel classification: A least-squares formulation, extensions, and analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **33**, 194 –200. 70

TAI, Y.C. & SPEED, T.P. (2006). A multivariate empirical Bayes statistic for replicated microarray time course data. *The Annals of Statistics*, **34**, 2387–2412. 7

TIBSHIRANI, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288. 4

TIPPING, M.E. & BISHOP, C.M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, **61**, 611–622. 38, 53, 55

TIPPING, M.E. & LAWRENCE, N.D. (2005). Variational inference for Student-t models: Robust Bayesian interpolation and generalised component analysis. *Neurocomputing*, **69**, 123–141. 10, 34

TITSIAS, M.K. & LAWRENCE, N.D. (2010). Bayesian Gaussian process latent variable model. In Y.W. Teh & D.M. Titterington, eds., *Proceedings of the Thirteenth International Workshop on Artificial Intelligence and Statistics*, vol. 9, 844–851, JMLR W&CP 9, Chia Laguna Resort, Sardinia, Italy. 44

TSILIGKARIDIS, T., HERO, A. & ZHOU, S. (2013). On convergence of kronecker graphical lasso algorithms. *Signal Processing, IEEE Transactions on*, **PP**, 1. 101

TUCKER, L. (1958). An inter-battery method of factor analysis. *Psychometrika*, **23**, 111–136. 3, 45, 50, 67, 74

VANHATALO, J., JYLÄNKI, P. & VEHTARI, A. (2009). Gaussian process regression with Student-t likelihood. In Y. Bengio, D. Schuurmans, J. Lafferty, C.K.I. Williams & A. Culotta, eds., *Advances in Neural Information Processing Systems 22*, 1910–1918. 10, 34

VIRTANEN, S., KLAMI, A. & KASKI, S. (2011). Bayesian CCA via group sparsity. In L. Getoor & T. Scheffer, eds., *Proceedings of the 28th International Conference on Machine Learning*, 457–464, ACM, New York, NY. 60, 75

Von Mises, R. (1964). Mathematical theory of probability and statistics. *Mathematical Theory of Probability and Statistics, New York: Academic Press, 1964*, **1**. 123

Wackernagel, H. (2003). *Multivariate geostatistics*. Springer. 104

Wang, C. (2007). Variational Bayesian approach to canonical correlation analysis. *Neural Networks, IEEE Transactions on*, **18**, 905–910. 60

Wegman, E. (1988). Reproducing kernel Hilbert spaces. *Encyclopedia of Statistical Sciences*. 17

Yuan, M. (2006). Flexible temporal expression profile modelling using the Gaussian process. *Computational statistics & data analysis*, **51**, 1754–1764. 10, 22

Zellner, A. (1962). An efficient method of estimating seemingly unrelated regressions and tests for aggregation bias. *Journal of the American statistical Association*, **57**, 348–368. 104

Zhang, Y. & Schneider, J. (2010). Learning multiple tasks with a sparse matrix-normal penalty. *Advances in Neural Information Processing Systems*, **23**, 2550–2558. 101