Aalto University
School of Science
Degree Programme of Engineering Physics and Mathematics

Eemeli Leppäaho

Transfer Learning with Group Factor Analysis

Master's Thesis
Espoo, January 16, 2013

Supervisor: Professor Samuel Kaski
Instructor: Arto Klami, D.Sc. (Tech.)

Aalto University
School of Science
Degree Programme of Engineering Physics and Mathematics

ABSTRACT OF
MASTER'S THESIS

Author: Eemeli Leppäaho

Title:
Transfer Learning with Group Factor Analysis

Date: January 16, 2013 **Pages:** viii + 53

Professorship: Information science **Code:** T-61

Supervisor: Professor Samuel Kaski

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

Modern measuring techniques allow us to get more and more data in less time and cheaper price. When analyzing data, one sample might be the gene expression of a cell or the activity of a human brain at a certain time, consisting of tens of thousands of features. Often we have much fewer samples than features, and simple methods will overfit the data. Factor models are designed to model this kind of high-dimensional data via a lower dimensional factor space. Factor analysis is the simplest factor model: it reconstructs each feature in the data as a weighted sum of the hidden factors (*components*).

In this thesis I examine group factor analysis (GFA), which is an extension of factor analysis for multiple data sets. High-dimensional data can often be naturally divided to different groups (*views*), which GFA uses as prior information by inferring the component activities for views instead of single features. This property combined with an automatic system for the component activity determination results in a powerful factor model.

In this thesis, GFA is extended to explicitly model hidden relations between different data views. This is done by generating their component activity matrix in two alternative ways: as samples of a multivariate normal distribution and as a product of two low-rank matrices. Both the extensions are solved via variational Bayesian inference, and are shown to model data with accuracy comparable to GFA. For data with many views low-rank GFA is the most accurate model.

Additionally the problem of small number of samples is dealt with two transfer learning setups: one being able to take advantage of background data with samples or features shared with target data, and the other introducing a novel transfer learning setup. It is shown, using both artificial and real data, that both of these setups allow us to form a better model when suitable background data is available. The real data consists of drug response profiles measured on cell lines using two different microarray platforms.

Keywords: Bayesian data analysis, factor models, transfer learning, variational inference

Language: English

Aalto-yliopisto
Perustieteiden korkeakoulu
Teknillisen fysiikan ja matematiikan tutkinto-ohjelma

DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Eemeli Leppäaho		
Työn nimi:	Siirto-oppimista ryhmäfaktorianalyysilla		
Päiväys:	16. tammikuuta 2013	**Sivumäärä:**	viii + 53
Professuuri:	Informaatiotekniikka	**Koodi:**	T-61
Valvoja:	Professori Samuel Kaski		
Ohjaaja:	Tekniikan tohtori Arto Klami		

Modernien mittaustekniikoiden avulla saadaan nykyään entistä enemmän aineistoa tutkittavaksi lyhyemmässä ajassa ja halvemmalla. Kun tutkimuksen kohteena ovat esimerkiksi solun geenien ilmentymisarvot tai ihmisaivojen toiminta, yksi näyte voi koostua kymmenistätuhansista muuttujista. Usein näytteitä on paljon vähemmän kuin muuttujia, jolloin yksinkertaiset menetelmät ylisovittuvat aineistoon. Faktorimallit on suunniteltu mallintamaan tällaista korkeaulotteista dataa matalaulotteisemman faktoriavaruuden avulla. Faktorianalyysi on näistä malleista yksinkertaisin: se rekonstruoi jokaisen aineiston muuttujan latenttien faktorien (*komponenttien*) painotettuna summana.

Tässä diplomityössä sovelletaan ja edelleenkehitetään ryhmäfaktorianalyysiä (GFA), joka on faktorianalyysin laajennus useille aineistojoukoille. Korkeaulotteinen data voidaan usein jakaa ryhmiin (*näkymiin*), jotka GFA ottaa huomioon mallintamalla komponenttiaktiivisuudet ryhmille yksittäisten muuttujien sijaan. Mallissa on myös mukana komponenttien relevanssin määrittävä osa. Nämä seikat tekevät GFA:sta käytännöllisen faktorimallin.

Tässä työssä laajennetaan ryhmäfaktorianalyysiä mallintamaan aineiston eri näkymien suhteita eksplisiittisesti. Tämä tehdään mallintamalla näkymien komponenttiaktiivisuudet kahdella vaihtoehtoisella tavalla: moniulotteisen normaalijakauman näytteinä sekä kahden matalan rangin matriisin tulona. Molemmat laajennukset ratkaistaan variationaalisen Bayes-päättelyn avulla, ja niiden tarkkuus aineiston mallintamisessa vastaa GFA:n tarkkuutta. Aineistossa, jossa on useita näkymiä, matalan rangin GFA on tarkin malli.

Pienen näytemäärän ongelmaan puututaan lisäksi kahdella siirto-oppimismenetelmällä. Toisessa hyödynnetään taustadataa, jossa on kohdedatan kanssa jaettuja näytteitä tai muuttujia. Toisessa lähestymistavassa on menetelmänä syvemmän tason siirto-oppiminen. Työssä osoitetaan sekä keinotekoisella että oikealla aineistolla, että molemmat menetelmät parantavat lopullista mallia, kunhan sopivaa taustadataa on saatavilla. Oikea aineisto koostuu solulinjoille mikrosiruilla tehdyistä lääkevastemittauksista.

Asiasanat:	bayesiläinen data-analyysi, faktorimallit, siirto-oppiminen, variationaalinen Bayes-päättely
Kieli:	Englanti

Acknowledgements

I appreciate being able to work on this thesis and associated reasearch in Statistical Machine Learning and Bioinformatics group, which is part of the Finnish Centre of Excellence in Computational Inference Research (COIN) and Helsinki Institute for Information Technology HIIT, at Department of Information and Computer Science at Aalto University School of Science. My research was funded by Aalto projects Brian and aivoAalto, and the computational simulations presented in this thesis were performed using computer resources within the Aalto University School of Science "Science-IT" project.

I wish to thank Arto Klami, Seppo Virtanen and Samuel Kaski for instructing me with the research that led to the models presented in this thesis. I am also grateful for Arto Klami for instructing my thesis and for Samuel Kaski supervising it. Suleiman Ali Khan deserves a special mention for providing me the drug response data, and helping me understand it. In addition, I would like to thank my whole research group for the past years and the ones to come.

Finally, I would like to thank Sini for proofreading this thesis, and for everything else.

Espoo, January 16, 2013

Eemeli Leppäaho

Symbols and Abbreviations

Scalars, vectors and matrices

Scalars are marked with lower and upper case symbols. Vectors are treated as column vectors and marked with boldface lowercase symbols. Matrices are boldface uppercase symbols. Transpose is marked with the symbol $^\top$. Column vectors corresponding to the ith row and the jth column of matrix $\mathbf{X}$ are marked with $\mathbf{X}_i$ and $\mathbf{X}_{\cdot j}$, respectively. Lists of matrices are marked with boldface uppercase symbols, such that $\mathbf{X}^m$ is the mth matrix in the list $\mathbf{X}$.

List of symbols

$\lvert\mathbf{A}\rvert$	Determinant of matrix $\mathbf{A}$
$\mathbb{E}[\mathbf{x}\lvert\mathbf{y}], \langle\mathbf{x}\rangle_\mathbf{y}$	Expectation of $\mathbf{x}$, given $\mathbf{y}$
$\mathbf{I}_k$	Identity matrix with k rows and columns
$\hat{\mathbf{x}}$	Prior for $\mathbf{x}$
$D_{KL}(q\lvert\lvert p)$	Kullback-Leibler divergence between distributions p and q
Γ	Gamma function
ψ	Digamma function
$\mathcal{L}$	Lower bound of a function
$\boldsymbol{\theta}$	Model parameters
$\mathcal{G}(x\lvert a, b)$	Gamma probability density at point x with shape and rate parameters a and b
$\mathcal{N}(\mathbf{x}\lvert\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian probability density at point $\mathbf{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$
$\mathcal{W}(\boldsymbol{\Lambda}\lvert\mathbf{V}, v)$	Wishart probability density at $\boldsymbol{\Lambda}$ with scale matrix $\mathbf{V}$ and v degrees of freedom

List of abbreviations

ARD	Automatic Relevance Determination
CCA	Canonical Correlation Analysis
CGFA	Correlated-Group Factor Analysis
DNA	Deoxyribonucleic Acid
FA	Factor Analysis
GFA	Group Factor Analysis
KL-divergence	Kullback-Leibler divergence
L-BFGS-B	Limited memory Broyden-Fletcher-Goldfarb-Shanno method with Box constraints
LRGFA	Low-Rank Group Factor Analysis
PCA	Principal Component Analysis
PCCA	Probabilistic Canonical Correlation Analysis
RNA	Ribonucleic Acid

Contents

Chapter 1

Introduction

1.1 Overview

In scientific research, there is at the moment more data available than ever. Whether one data sample is a 10000-dimensional vector of gene expression values measured as a response for a certain drug or the blood-oxygen-level-dependent in different human brain regions at a certain time, there is a clear need for sophisticated statistical methods in modeling the data [22][25]. We often face data where the amount of information per sample is much higher that the total number of samples. This is called the "small n, large p" problem. Additionally, a large part of the measurements might actually be irrelevant noise.

The "small n, large p" problem is often dealt with dimensionality reduction, that is, finding a representation of data where we have significantly fewer features. There is a wide variety of dimensionality reduction methods: In the simplest case one might have prior information about relevant features, or the features might be chosen by some ad-hoc criteria, such as picking the features with most variance [23]. Another approach is to build a model that lets data decide which features are relevant. This too can be done in many ways, and in this thesis we consider factor models. Probably the most well-known factor model is factor analysis, which is used to explain p possibly correlated variables with $k \ll p$ uncorrelated latent factors (or components) [32].

In this thesis we examine the model *group factor analysis* (GFA), which can be viewed as a generalization of factor analysis for multiple data sets [33]. In high-dimensional data, there often exists some sort of natural grouping of variables. The grouping might present different pathways in gene expression data or different brain regions of fMRI data. In this thesis the term *view* is

used to describe these different groups, reflecting the idea that we measure a shared thing from different perspectives. In group factor analysis these views are taken into account in the latent space: all variables in one view share the same components, but different views may share different components. Thus, when determining the component activities, it is enough to infer MK parameters instead of DK, where M is the number of views, K is the number of components and D is the total dimensionality.

For huge data sets with many views, inferring MK independent component activity parameters might still not be optimal. With large M surely there are some views that have much in common or some that have very little. This is the idea behind the two GFA modifications developed in this thesis: one where the activities are sampled from a normal distribution and the other with two low-rank matrices forming them. In vanilla GFA two views might have identical component activities, but in these modifications this kind of similarity is explicitly modeled.

Increased experimental data size has to be taken into account when designing models. Besides that, there are more and more open databases where researchers all over the world have uploaded data from their experiments. For example European Bioinformatics Institute maintains roughly 40 biological databases.[1] Given similar measurements, this kind of background data could be used for example to help dealing with the "small n, large p" problem. If someone has measured the effects of one thousand drugs on cell lines, why not include them to the model of our ten drugs? In the simplest scenario we can just add some samples or views to our own data. In general, we are facing a transfer learning problem: how to use the background information to get the best possible model for target data? In this thesis the question is answered by developing a sequential Bayesian learning GFA and a novel way of transfer learning: transferring the view correlation structure from a background data.

1.2 Structure of the thesis

The structure of the thesis is as follows: basic mathematical methods and the application area are reviewed in chapter 2. On the methodological side, Bayesian inference is explained first along with the special case of variational inference. They are followed by an introduction to transfer learning and factor models, and finally by introducing the biological background.

Chapter 3 begins with an introduction to group factor analysis and deriva-

[1] `http://www.ebi.ac.uk/Databases/`

tion of its inference equations. After that, as the first main contribution of this thesis, two novel extensions of GFA are derived: correlated-group factor analysis and low-rank factor analysis.

Two new transfer learning extensions for GFA are presented in chapter 4. Different types of prior-inducing schemes are discussed and two transfer learning extensions are derived: traditional sequential Bayesian learning model and a hierarchical model for transferring view correlation structure.

Chapter 5 contains experiments on simulated data for all the GFA models presented in this thesis. The purpose of these experiments is to show that the implementations work in the way they are designed to.

In chapter 6, experiments with real drug response data are carried out to see how applicable the models are. Independently learning the target data is compared to a transfer learning approach, where additional information is extracted from the background data.

The thesis is concluded in chapter 7, where the models and experimental results are discussed in-depth.

1.3 Contributions of the thesis

The main contributions of this thesis are the extensions of the GFA model: correlated-group factor analysis, low-rank factor analysis and transfer learning models. These were formulated in collaboration with Arto Klami, Seppo Virtanen and Samuel Kaski, who presented GFA as a factor analysis model generalized for multiple data sets [33]. The author derived and implemented the extensions, performed the experiments and wrote the thesis independently.

Chapter 2

Background

This chapter begins with an introduction to Bayesian statistics, including a specific methodology: variational Bayesian inference. Also the general framework of transfer learning and factor models are presented. Finally, we discuss the biological background of drug response experiments, since it is necessary to know what type of data we are modelling.

When introducing Bayesian inference we use $\mathbf{x}$ to denote the data and do not specify its nature in detail. This is because the inference methods are very general, and even if the data were discrete, the only change one would have to make is to replace the integrations with summations. While dealing with factor models we denote the data with $\mathbf{X} \in \mathbb{R}^{D \times N}$, meaning that the data consist of N samples for which we have D continuous measurements.

2.1 Bayesian inference

In traditional statistics we are often interested in finding parameters $\boldsymbol{\theta}$ that maximize the likelihood of data $\mathbf{x}$:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(\mathbf{x}|\boldsymbol{\theta}), \tag{2.1}$$

where $\hat{\boldsymbol{\theta}}$ is called the maximum likelihood estimate. In Bayesian statistics the parameters have a prior $p(\boldsymbol{\theta})$, allowing us to infer a probability distribution for the parameters. This is done using Bayes' rule:

$$p(\boldsymbol{\theta}|\mathbf{x}) = \frac{p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{x})} \propto p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}). \tag{2.2}$$

We are interested in the posterior distribution of parameters, $p(\boldsymbol{\theta}|\mathbf{x})$, given data and prior $p(\boldsymbol{\theta})$. Probability of the data, $p(\mathbf{x})$, is constant with

respect to the model parameters and can thus be ignored in many inference tasks.

In Bayesian inference the model is defined by the likelihood and the prior. The prior is usually chosen from a distribution family that is conjugate to the likelihood $p(\mathbf{x}|\boldsymbol{\theta})$, assuring that the posterior is a closed form distribution too. In the so-called objective Bayesian setup the prior is chosen such that it contains as little information as possible; an uninformative prior results in the traditional maximum likelihood estimate, with the exception that we get a posterior distribution instead of a point. More generally, which is often referred as the subjective setup, the prior actually reflects prior beliefs.

If either the likelihood $p(\mathbf{x}|\boldsymbol{\theta})$ or the prior $p(\boldsymbol{\theta})$ is a complex distribution, the posterior $p(\boldsymbol{\theta}|\mathbf{x})$ is usually not tractable. In that case it needs to be approximated.

2.1.1 Variational Bayesian inference

In this section we describe a framework for approximating the posterior of the model parameters, $p(\boldsymbol{\theta}|\mathbf{x})$, with another distribution $q(\boldsymbol{\theta})$. Following Bishop [5], we start by formulating the marginal distribution $p(\mathbf{x})$ as:

$$\begin{aligned}
\ln p(\mathbf{x}) &= \int q(\boldsymbol{\theta}) \ln p(\mathbf{x}) d\boldsymbol{\theta} && (2.3)\\
&= \int q(\boldsymbol{\theta}) \ln \frac{p(\boldsymbol{\theta}|\mathbf{x})p(\mathbf{x})}{p(\boldsymbol{\theta}|\mathbf{x})} d\boldsymbol{\theta} && (2.4)\\
&= \int q(\boldsymbol{\theta}) \ln p(\mathbf{x}, \boldsymbol{\theta}) d\boldsymbol{\theta} - \int q(\boldsymbol{\theta}) \ln p(\boldsymbol{\theta}|\mathbf{x}) d\boldsymbol{\theta} && (2.5)\\
&= \int q(\boldsymbol{\theta}) \ln \frac{p(\mathbf{x}, \boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta} - \int q(\boldsymbol{\theta}) \ln \frac{p(\boldsymbol{\theta}|\mathbf{x})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta} && (2.6)\\
&:= \mathcal{L}(q) + D_{KL}(q||p), && (2.7)
\end{aligned}$$

where in equation (2.3) the logarithm of the marginal likelihood is formulated by integrating over the variational distribution $q(\boldsymbol{\theta})$. Bayes' rule is applied in equation (2.4), after which the division in the logarithm is changed to a subtraction of two logarithms, and the conditional probability is changed to a joint probability in equation (2.5). In equation (2.6) we have added $\int q(\boldsymbol{\theta}) \ln q(\boldsymbol{\theta}) d\boldsymbol{\theta} - \int q(\boldsymbol{\theta}) \ln q(\boldsymbol{\theta}) d\boldsymbol{\theta}$ and combined it with the existing integrands. Finally, equation (2.6) is divided in two terms that are interpreted as the lower bound $\mathcal{L}(q)$ and the KL-divergence $D_{KL}(q||p)$ [21].

The derivation in equations (2.3)-(2.7) holds for all types of $q(\boldsymbol{\theta})$. In equation (2.7) the KL-divergence measures the distance of the true posterior and variational distribution $q(\boldsymbol{\theta})$, so we would like to minimize it. As the

marginal distribution $p(\mathbf{x})$ is a constant, this can be done equivalently by maximizing $\mathcal{L}(q)$. Since $D_{KL}(q||p) \geq 0$, $\mathcal{L}(q)$ is a lower bound for $\ln p(\mathbf{x})$. The equality applies only when $q(\boldsymbol{\theta})$ matches the true posterior.

In order to learn a tractable $q(\boldsymbol{\theta})$, we will follow a framework called *mean field theory* [30], where the variational distribution $q(\boldsymbol{\theta})$ is factorized with respect to all parameters:

$$q(\boldsymbol{\theta}) = \prod_i q_i(\boldsymbol{\theta}_i). \tag{2.8}$$

We use shorthand notation q_i to denote $q_i(\boldsymbol{\theta}_i)$. Given the factorized variational distribution, the lower bound becomes

$$\mathcal{L}(q) = \int \prod_i q_i \left(\ln p(\mathbf{x}, \boldsymbol{\theta}) - \sum_i \ln q_i \right) d\theta \tag{2.9}$$

$$= \int q_j \left(\int \ln p(\mathbf{x}, \boldsymbol{\theta}) \prod_{i \neq j} q_i d\boldsymbol{\theta}_i \right) d\boldsymbol{\theta}_j - \int q_j \ln q_j d\boldsymbol{\theta}_j + \text{const} \tag{2.10}$$

$$= \int q_j \ln \tilde{p}(\mathbf{x}, \boldsymbol{\theta}_j) d\boldsymbol{\theta}_j - \int q_j \ln q_j d\boldsymbol{\theta}_j + \text{const} \tag{2.11}$$

$$= \int q_j \ln \frac{\tilde{p}(\mathbf{x}, \boldsymbol{\theta}_j)}{q_j} d\boldsymbol{\theta}_j + \text{const}, \tag{2.12}$$

where

$$\ln \tilde{p}(\mathbf{x}, \boldsymbol{\theta}_j) := \int \ln p(\mathbf{x}, \boldsymbol{\theta}) \prod_{i \neq j} q_i d\boldsymbol{\theta}_i = \mathbb{E}[\ln p(\mathbf{x}, \boldsymbol{\theta}) | q_{i \neq j}], \tag{2.13}$$

and the latter part of equation (2.11) is the entropy of q_j, denoted by $H(q_j)$. Equation (2.12) is actually the negative KL-divergence of q_j and $\tilde{p}(\mathbf{x}, \boldsymbol{\theta}_j)$, so the maximum is

$$\mathbb{E}[\ln p(\mathbf{x}, \boldsymbol{\theta}) | q_{i \neq j}] + \text{const}. \tag{2.14}$$

This results in an iterative mean field variational Bayesian algorithm, as presented in algorithm 1. The iteration is needed since the optimum of $q_i(\boldsymbol{\theta}_i)$ depends on the expectations of the other factors. Convergence to a local optimum is guaranteed because the bound is convex with respect to each of the factors $q_i(\boldsymbol{\theta}_i)$ [7]. Algorithm 1 is closely related to expectation-maximization algorithms, as presented by Dempster [10].

If the optimum in equation (2.14) is not tractable, equation (2.12) has to be maximized some other way with respect to q_j. This can be done for example with numerical methods [6].

Algorithm 1: Mean field variational Bayesian inference

```
Start with some initial distribution for q(θ)
while not converged do
    for i=1,...,I do
        Update q_i(θ_i), while holding the rest of the variational
        distribution fixed
    end
end
```

2.2 Transfer learning

As discussed in the previous section, the goal of Bayesian inference is to find a posterior estimate $p(\boldsymbol{\theta}|\mathbf{x})$. If there are enough samples and our modeling assumptions are correct, the posterior is an accurate distribution for the parameters [11]. However, many experiments are expensive to run, and thus the sample size might be too small. If the dimensionality for each sample is high, we are dealing with the "small n, large p" -problem [34]. This problem can be dealt with the means of transfer learning, along with other approaches such as feature selection [15].

The data acquired from our experiments is called *target* data. We are interested in modeling it as well as possible. If there exists some related *background* data, we would like to transfer maximal amount of knowledge from the background data for modeling the target. Both the target and background are described by a domain [29]

$$\mathcal{D} = \{\mathcal{X}, P(\mathbf{X})\}, \tag{2.15}$$

where $\mathcal{X}$ is a feature space and $P(\mathbf{X})$ is the marginal probability distribution of data $\mathbf{X} = \{\mathbf{X}_1, ..., \mathbf{X}_N\} \in \mathcal{X}$. The target and background domains differ if they have different features or marginal distributions.

For identical domains transfer learning is trivial: background samples can be pooled together with the target, resulting in a more accurate model. Likewise, if the background and target share the same samples but have different domains, it is sufficient to combine them together. Non-trivial cases can be dealt with a general transfer learning algorithm: first by modeling background and then by using the relevant parts of the posterior as a prior for the target data. This is also known as sequential Bayesian learning, and in exact inference the posterior for target data will be the same as if it was modeled jointly with the background [14]. If the background model is known along with the data, sequential learning can offer significant computational benefit, since the background data does not need to be modeled any more.

Knowledge from shared features and samples can easily be transferred using sequential Bayesian learning. The same logic applies to all model parameters: if there is a reason to assume they are similar, the posterior of the background should be used as a prior for the target. These and other types of transfer learning schemes have been discussed by Pan and Yang in their overview on transfer learning [29].

2.3 Factor models

In many real-world applications we observe data with a huge dimensionality; for example in gene expression measurements there might be thousands of genes as variables. Often many of the variables are correlated, and thus we would like to present the same information in a different, lower dimensional space, aiming to capture information and leave out noise. Latent variable models attend to do this, and they are useful in preprocessing high-dimensional data and analyzing it as is. All the models studied in this thesis are factor models.

2.3.1 Factor analysis

Factor analysis is a statistical method that models observed variables with a smaller amount of *factors* [32]. Given zero mean data matrix $\mathbf{X}$, our model is:

$$\mathbf{X} = \mathbf{W}\mathbf{Z} + \boldsymbol{\epsilon}, \tag{2.16}$$

where $\mathbf{X} \in \mathbb{R}^{D\times N}, \mathbf{W} \in \mathbb{R}^{D\times K}, \mathbf{Z} \in \mathbb{R}^{K\times N}$ and $\boldsymbol{\epsilon} \in \mathbb{R}^{D\times N}$. The data matrix is presented via the lower dimensional factors $\mathbf{Z}$, and the loading matrix (projection matrix) $\mathbf{W}$ describes the linear relationships of the features. The error term $\boldsymbol{\epsilon}$ has zero mean and diagonal covariance $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, ..., \sigma_D)$. Additionally, we assume that the noise is independent of the factors, which have zero mean and identity covariance matrix. Thus,

$$\mathrm{cov}(\mathbf{X}) = \mathrm{cov}(\mathbf{W}\mathbf{Z} + \boldsymbol{\epsilon}) = \mathrm{cov}(\mathbf{W}\mathbf{Z}) + \boldsymbol{\Sigma} = \mathbf{W}\mathrm{cov}(\mathbf{Z})\mathbf{W}^\top + \boldsymbol{\Sigma} = \mathbf{W}\mathbf{W}^\top + \boldsymbol{\Sigma}, \tag{2.17}$$

and therefore factor analysis models all the correlations between the variables, leaving only diagonal noise with different variance for each variable. The projection matrix captures the correlation structure, whereas the factors represent the data in a space with no correlated variables.

Factor analysis is a widely used latent variable model for example in chemistry [26]. It is also closely related to principal component analysis, which has an isotropic noise covariance instead of just diagonal [18].

2.3.2 Probabilistic canonical correlation analysis

Canonical correlation analysis (CCA), as presented by Hotelling [17], is a way of finding maximal linear correlations between two sets of variables, namely $\mathbf{X}$ and $\mathbf{Y}$, with dimensionalities D_X and D_Y. The first pair of canonical correlations, $(\mathbf{a}^\top\mathbf{X}, \mathbf{b}^\top\mathbf{Y})$ is chosen to achieve maximal correlations. The next pair is again chosen in order to maximize correlation, with the restriction that it has to be uncorrelated with the preceding canonical variables. In the end, we get two sets of variables, $\mathbf{A}^\top\mathbf{X}$ and $\mathbf{B}^\top\mathbf{Y}$, that have a diagonal correlation matrix with descending elements.

Probabilistic canonical correlation analysis (PCCA) gives CCA a factor model interpretation [4]. The model is defined as follows:

$$p(\mathbf{Z}) \sim \prod_{i=1}^{N} \mathcal{N}(\mathbf{Z}_{\cdot i}|\mathbf{0}, \mathbf{I}_K) \tag{2.18}$$

$$p(\mathbf{X}|\mathbf{Z}) \sim \prod_{i=1}^{N} \mathcal{N}(\mathbf{X}_{\cdot i}|\mathbf{W}_X\mathbf{Z}_{\cdot i} + \boldsymbol{\mu}_X, \boldsymbol{\Sigma}_X) \tag{2.19}$$

$$p(\mathbf{Y}|\mathbf{Z}) \sim \prod_{i=1}^{N} \mathcal{N}(\mathbf{Y}_{\cdot i}|\mathbf{W}_Y\mathbf{Z}_{\cdot i} + \boldsymbol{\mu}_Y, \boldsymbol{\Sigma}_Y) \tag{2.20}$$

where K is the latent dimensionality. The $\mathbf{W} \in \mathbb{R}^{D\times K}$ are the projection matrices, with corresponding positive semi-definite covariance matrices $\boldsymbol{\Sigma} \in \mathbb{R}^{D\times D}$.

Given two data sets and latent dimensionality K, PCCA has a maximum likelihood solution [4]. Similar to standard CCA, the solution tries to model the correlations between the data sets, which allow the model to have more statistical strength. CCA and PCCA are standard methods for modeling two data sets, and have been used for example to learn a bilingual dictionary from two monolingual text corpora [16].

2.3.3 Model complexity control

A major drawback of factor analysis is that the number of components, and thus the dimensionality of the latent space has to be fixed. In this thesis this problem has been dealt with *automatic relevance determination* (ARD), originally formulated in the framework of neural networks [24][27]. For factor analysis it could be implemented by assigning the columns of loading matrix the distribution

$$\mathbf{W} \sim \prod_{k=1}^{K} \mathcal{N}(\mathbf{W}_{\cdot k}|\mathbf{0}, \frac{1}{\boldsymbol{\alpha}_k}\mathbf{I}_D), \tag{2.21}$$

where the $\boldsymbol{\alpha} = \{\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_K\}$ are called ARD parameters. If the data can be described via a k-dimensional latent space and we are modeling it with $K > k$ factors, the extra $K - k$ factors will have effectively zero variance. This happens since they do not aid in modeling the data significantly, but instead they do increase the likelihood of $\mathbf{W}$ when $\boldsymbol{\alpha}_k \to \infty$.

Automatic relevance determination has been applied in many Bayesian models. Nielsen [28] presented factor analysis with ARD parameters and solved it with variational inference methods.

2.4 Biological background

In this thesis our application comes from drug response experiments. Those experiments try to measure how various drugs affect human body. To have some idea of how drug effects could be measured, we need have some basic knowledge of the target object, human body.

The functional unit of every living organism is the cell. Humans have approximately ten trillion of them, all of which have one thing in common: DNA (*deoxyribonucleic acid*). DNA contains the information needed for the organism's functioning and developing, and this information is utilized by proteins that are the functional units inside the cell [1]. Messenger RNA (*ribonucleic acid*) forms a protein from a stretch of DNA (gene) in a process called gene expression.

When comparing patients having different diseases and treatments, we would expect to see different protein levels in their cells, since proteins are the basic functional units. Unfortunately this kind of information is very hard to acquire. There is, however, an efficient way to measure the next best thing. A DNA microarray is an array with fragments of DNA, called probes, that are matched with messenger RNA in cells. The probes can be mapped into genes, allowing us to infer the expression level for each gene.

2.4.1 Drug response experiments

The action mechanism of many drugs is enzyme inhibition. This means that they act in cells and decrease the activity of some enzymes (usually proteins) for example by binding into them. This effect can be seen in gene expression measurements, allowing us to infer the drug action mechanisms. [19]

Lamb et al. [22] published a gene expression database for drug response experiments. Instead of doing possibly harmful experiments on patients, they used isolated human cells and measured their gene expression values

with DNA microarrays. This type of data can be used to analyze connections among drugs, genes and diseases. The data contains measurements done with two different microarray platforms, both measuring approximately 22000 genes. The number of drugs measured with the platforms are 682 and 313. In both the platforms the genes can be divided into 217 different groups, corresponding to the pathways via which they interact. The pathways describe series of chemical reactions occurring within a cell.

Chapter 3

Group Factor Analysis

This chapter begins with an introduction to group factor analysis and derivation of variational Bayesian inference update equations for solving an approximate posterior of the model parameters. After that the limitations of GFA are discussed, and two new GFA extensions are presented to address them.

Group factor analysis can be thought of as an extension of factor analysis to multiple data sets that have independent sparsity inducing ARD parameters [33]. Whereas in factor analysis the data consists of N samples with dimensionality D each, now we are interested in M such data matrices, with varying dimensionality D_m. All the views are modeled using shared latent variables, as clarified in figure 3.1. The key idea is that we have a set of samples that can be presented via M different views that have some specific and some shared variation. An example of a real world data set was published by Lamb at al. [22]: there are N drugs tested on cell lines with different gene expression measurements as features, grouped into different pathways. We will return to this example in the experimental part of this thesis.

The generative model of GFA is:

$$p(\mathbf{X}|\mathbf{W}, \mathbf{Z}, \boldsymbol{\tau}) \sim \prod_{m=1}^{M} \prod_{i=1}^{N} \mathcal{N}(\mathbf{X}_{\cdot i}^{m} | \mathbf{W}^{m} \mathbf{Z}_{\cdot i}, \boldsymbol{\tau}_{m}^{-1} \mathbf{I}_{D_m}), \tag{3.1}$$

where $\mathbf{X}_{\cdot i}^{m}$ is the ith sample of view $\mathbf{X}^{m}$, $\mathbf{W}^{m}$ is the projection matrix for view m, $\mathbf{Z}_{\cdot i}$ is the latent representation of the ith sample and $\boldsymbol{\tau}_m$ is the noise

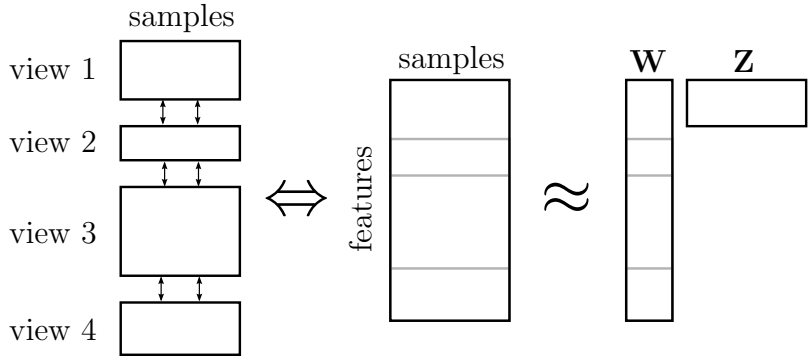


Figure 3.1: Given data where the samples are described by several views, factor analysis can be used to model it by concatenating the views. This results in a features×samples-matrix, which is modeled as a product of two matrices: projection matrix $\mathbf{W}$ and latent components $\mathbf{Z}$. GFA takes the prior grouping into account by having a separate projection matrix for each group (gray lines). In our application different drugs are samples, genes are features and pathways are views.

precision. The priors of the model are defined as

$$p(\mathbf{Z}) \sim \prod_{i=1}^{N} \mathcal{N}(\mathbf{Z}_{\cdot i}|\mathbf{0}, \mathbf{I}_K) \tag{3.2}$$

$$p(\boldsymbol{\tau}|a^\tau, b^\tau) \sim \prod_{m=1}^{M} \mathcal{G}(\boldsymbol{\tau}_m|a^\tau, b^\tau) \tag{3.3}$$

$$p(\mathbf{W}|\boldsymbol{\alpha}) \sim \prod_{m=1}^{M}\prod_{k=1}^{K} \mathcal{N}(\mathbf{W}^m_{\cdot k}|\mathbf{0}, \frac{1}{\boldsymbol{\alpha}_{mk}}\mathbf{I}_{D_m}) \tag{3.4}$$

$$p(\boldsymbol{\alpha}|a^\alpha, b^\alpha) \sim \prod_{m=1}^{M}\prod_{k=1}^{K} \mathcal{G}(\boldsymbol{\alpha}_{mk}|a^\alpha, b^\alpha), \tag{3.5}$$

where $\boldsymbol{\alpha}_{mk}$ is an ARD parameter controlling the variance of component k for view m. Gamma-priors for both $\boldsymbol{\tau}$ and $\boldsymbol{\alpha}$ will be chosen to be uninformative, so the a and b parameters for both are fixed to 10^{-14}. Thus the prior expected values of $\boldsymbol{\tau}$ and $\boldsymbol{\alpha}$ are 1 and their prior variances are 10^{14}. The data and parameter dimensions are $\mathbf{X}^m \in \mathbb{R}^{D_m \times N}$, $\mathbf{W}^m \in \mathbb{R}^{D_m \times K}$, $\mathbf{Z} \in \mathbb{R}^{K \times N}$, $\boldsymbol{\alpha} \in \mathbb{R}^{M \times K}$ and $\boldsymbol{\tau} \in \mathbb{R}^M$, where D_m is the dimensionality of view m and K is the number of latent components. The plate model presentation of GFA is in figure 3.2.

Similar to any other factor model, GFA models correlated variables via an uncorrelated lower dimensional latent space. The relations of the variables

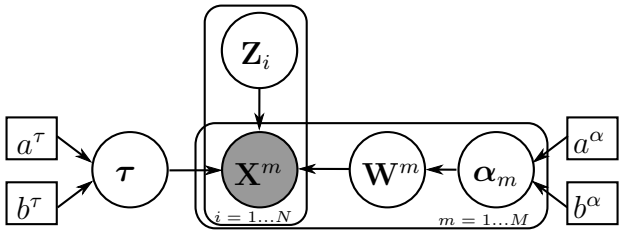


Figure 3.2: Plate model of group factor analysis. Gray node presents observed variables. Rounded plates denote random variables and rectangles fixed parameters.

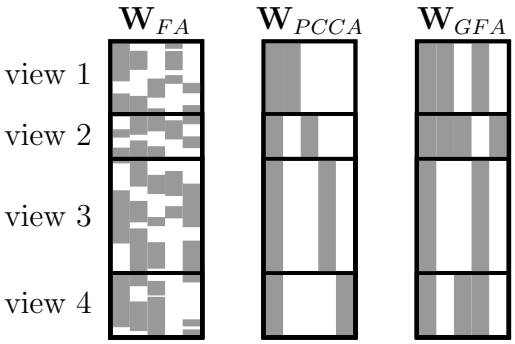


Figure 3.3: White parts of the projection matrices denote elements close to zero, not affecting the factor model. Gray parts denote elements deviating significantly from zero. Factor analysis does not take into account the grouping of features and determines the activity of each component separately for each feature. PCCA extended to more than two views takes the feature grouping into account, but allows only components that are shared with all the views or specific to just one view. GFA allows the views to share the components arbitrarily.

are taken into account in the sparse projection matrix $\mathbf{W}$, as two strongly correlated variables are likely to have the same active components. However, the structure of the ARD matrix $\boldsymbol{\alpha}$ is a feature that distinguishes GFA from other factor models. Factor analysis applied to pooled data sets infers the activity of each component independently for each feature, requiring DK free parameters in total, where $D = \sum_{m=1}^{M} D_m$ [28]. When the data has a high dimensionality, it is more reasonable to take the views into account in order to reduce the number of free parameters. GFA does this by inferring the component activity independently for each view, thus requiring MK parameters. The independence allows a single component to be shared between an arbitrary subset of all the views. This differs from PCCA generalized to

many views, where only components shared between all the views or ones specific to one view only are allowed [4]. The different component activity structure can be seen in figure 3.3, as visualized in the projection matrices of these factor models.

The set of all GFA parameters shall be denoted as $\boldsymbol{\theta} = \{\mathbf{Z}, \mathbf{W}, \boldsymbol{\alpha}, \boldsymbol{\tau}\}$. Their posterior can be computed using Bayes' rule:

$$
\begin{aligned}
& p(\boldsymbol{\theta}|\mathbf{X}) && (3.6)\\
=& p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})/p(\mathbf{X}) && (3.7)\\
=& p(\mathbf{X}|\mathbf{Z},\mathbf{W},\boldsymbol{\alpha},\boldsymbol{\tau})p(\mathbf{Z})p(\mathbf{W}|\boldsymbol{\alpha})p(\boldsymbol{\alpha}|a^\alpha,b^\alpha)p(\boldsymbol{\tau}|a^\tau,b^\tau)/p(\mathbf{X}) && (3.8)\\
=& \prod_{i=1}^{N} e^{-\frac{1}{2}\mathbf{Z}_{\cdot i}^\top \mathbf{Z}_{\cdot i}} \prod_{m=1}^{M} \left[\prod_{i=1}^{N} \left(|\boldsymbol{\tau}_m \mathbf{I}_{D_m}|^{\frac{1}{2}} e^{-\frac{\boldsymbol{\tau}_m}{2}(\mathbf{X}_{\cdot i}^m - \mathbf{W}^m \mathbf{Z}_{\cdot i})^\top (\mathbf{X}_{\cdot i}^m - \mathbf{W}^m \mathbf{Z}_{\cdot i})} \right) \right. \\
& \left. \prod_{k=1}^{K} \left(\boldsymbol{\alpha}_{mk}^{\frac{D_m}{2}} e^{-\frac{1}{2}\boldsymbol{\alpha}_{mk}\mathbf{W}_{\cdot k}^{m\top}\mathbf{W}_{\cdot k}^m} \right) \prod_{k=1}^{K} \boldsymbol{\alpha}_{mk}^{a^\alpha - 1} e^{-b^\alpha \boldsymbol{\alpha}_{mk}} \boldsymbol{\tau}_m^{a^\tau - 1} e^{-b^\tau \boldsymbol{\tau}_m} \right] \times \text{const.} && (3.9)
\end{aligned}
$$

The posterior in equation (3.9) can be split into two parts: the prior of the latent variables $\mathbf{Z}$ is independent of all the other variables, and the rest of the posterior is a product over all the views. Thus the complete log-likelihood for view m is:

$$
\begin{aligned}
\mathcal{L}_m(\mathbf{X},\boldsymbol{\theta}) =& \frac{D_m N}{2} \ln \boldsymbol{\tau}_m - \frac{\boldsymbol{\tau}_m}{2} \sum_{i=1}^{N} (\mathbf{X}_{\cdot i}^m - \mathbf{W}^m \mathbf{Z}_{\cdot i})^\top (\mathbf{X}_{\cdot i}^m - \mathbf{W}^m \mathbf{Z}_{\cdot i}) \\
& + \frac{D_m}{2} \sum_{k=1}^{K} \ln \boldsymbol{\alpha}_{mk} - \frac{1}{2} \sum_{k=1}^{K} \boldsymbol{\alpha}_{mk} \mathbf{W}_{\cdot k}^{m\top} \mathbf{W}_{\cdot k}^m \\
& + \sum_{k=1}^{K} (a^\alpha - 1) \ln \boldsymbol{\alpha}_{mk} - \sum_{k=1}^{K} b^\alpha \boldsymbol{\alpha}_{mk} \\
& + (a^\tau - 1) \ln \boldsymbol{\tau}_m - b^\tau \boldsymbol{\tau}_m + \text{const.} \qquad (3.10)
\end{aligned}
$$

The complete log-likelihood is a sum of (3.10) over views $m = 1, ..., M$, plus a view-independent expression

$$
\mathcal{L}(\mathbf{Z}) = -\frac{1}{2} \sum_{i=1}^{N} \mathbf{Z}_{\cdot i}^\top \mathbf{Z}_{\cdot i}. \qquad (3.11)
$$

3.1 Update equations for variational Bayesian inference

The posterior of model parameters in equation (3.9) is a complicated formula and thus cannot be solved in closed form. Here we will apply mean-field variational Bayesian inference presented in section 2.1.1. The posterior distribution is approximated as:

$$p(\boldsymbol{\theta}|\mathbf{X}) \approx q(\boldsymbol{\theta}) = q(\mathbf{Z})q(\mathbf{W})q(\boldsymbol{\alpha})q(\boldsymbol{\tau}), \tag{3.12}$$

which is the key assumption of mean-field variational Bayesian inference. The variational distribution $q(\boldsymbol{\theta})$ can be solved using algorithm 1, for which we need to compute the conditional distribution updates. This is done in the following section, where the following notation is used for simplicity: the complete log-likelihood as a function of parameter $\boldsymbol{\phi}$ is denoted as $\mathcal{L}(\boldsymbol{\phi})$ and its expectation given the other variational distributions by $\langle \mathcal{L}(\boldsymbol{\phi}) \rangle_{q(\boldsymbol{\theta})}$. Likewise, the log-likelihood for view m is denoted by $\mathcal{L}_m$. We can further factorize $q(\boldsymbol{\theta})$ with respect to views, since the complete log-likelihood is a sum over them. For clarity, all the update equations are displayed in frames. Additionally, a slightly different notation is adopted to separate the distributional parameters from actual model parameters: the superscript of a distributional parameter is the related model parameter, and subscripts will be used more than one distributional parameter is needed. For example $\boldsymbol{\Sigma}_m^W$ is a parameter of $q(\mathbf{W}^m)$.

3.1.1 The $q(\mathbf{Z})$ distribution

The complete log-likelihood as a function of $\mathbf{Z}$ is

$$\mathcal{L}(\mathbf{Z}) = \sum_{i=1}^{N} \left[-\frac{1}{2}\mathbf{Z}_{\cdot i}^\top \mathbf{Z}_{\cdot i} - \frac{1}{2}\sum_{m=1}^{M} \boldsymbol{\tau}_m (\mathbf{X}_{\cdot i}^m - \mathbf{W}^m \mathbf{Z}_{\cdot i})^\top (\mathbf{X}_{\cdot i}^m - \mathbf{W}^m \mathbf{Z}_{\cdot i}) \right], \tag{3.13}$$

and its expectation given the other variational distributions is

$$\sum_{i=1}^{N} \left[\mathbf{Z}_{\cdot i}^\top \sum_{m=1}^{M} \langle \mathbf{W}^m \rangle \langle \boldsymbol{\tau}_m \rangle \mathbf{X}_{\cdot i}^m - \frac{1}{2}\mathbf{Z}_{\cdot i}^\top \left(\mathbf{I}_k + \sum_{m=1}^{M} \langle \boldsymbol{\tau}_m \rangle \langle \mathbf{W}^{m\top} \mathbf{W}^m \rangle \right) \mathbf{Z}_{\cdot i} \right]. \tag{3.14}$$

Equation (3.14) is the optimal log-density for $q(\mathbf{Z})$, given the other parameters. By exponentiating the log-density, we can see that the optimal

$q(\mathbf{Z})$ is a normal distribution,

$$q(\mathbf{Z}) = \prod_{i=1}^{N} q(\mathbf{Z}_{\cdot i}) = \prod_{i=1}^{N} \mathcal{N}(\mathbf{Z}_{\cdot i} | \mathbf{M}_i^Z, \mathbf{\Sigma}^Z), \tag{3.15}$$

where the parameters are:

$$\mathbf{\Sigma}^Z = \left(\mathbf{I}_k + \sum_{m=1}^{M} \langle \boldsymbol{\tau}_m \rangle \langle \mathbf{W}^{m\top} \mathbf{W}^m \rangle \right)^{-1} \tag{3.16}$$

$$\mathbf{M}_i^Z = \sum_{m=1}^{M} \mathbf{\Sigma}^Z \langle \mathbf{W}^m \rangle^\top \langle \boldsymbol{\tau}_m \rangle \mathbf{X}_{\cdot i}^m. \tag{3.17}$$

3.1.2 The $q(\mathbf{W}^m)$ distribution

The parts of equation (3.10) depending on $\mathbf{W}^m$ are:

$$-\frac{1}{2} \sum_{i=1}^{N} \boldsymbol{\tau}_m (\mathbf{X}_{\cdot i}^m - \mathbf{W}^m \mathbf{Z}_{\cdot i})^\top (\mathbf{X}_{\cdot i}^m - \mathbf{W}^m \mathbf{Z}_{\cdot i}) - \frac{1}{2} \sum_{k=1}^{K} \boldsymbol{\alpha}_{mk} \mathbf{W}_{\cdot k}^{m\top} \mathbf{W}_{\cdot k}^m. \tag{3.18}$$

The part of equation (3.18) not depending on $\mathbf{X}_{\cdot i}^m$ is

$$-\frac{\boldsymbol{\tau}_m}{2} \sum_{i=1}^{N} (\mathbf{W}^m \mathbf{Z}_{\cdot i})^\top \mathbf{W}^m \mathbf{Z}_{\cdot i} = -\frac{\boldsymbol{\tau}_m}{2} \sum_{i=1}^{N} \sum_{j=1}^{D_m} (\mathbf{W}_j^{m\top} \mathbf{Z}_{\cdot i})^\top \mathbf{W}_j^{m\top} \mathbf{Z}_{\cdot i} \tag{3.19}$$

$$= \boldsymbol{\tau}_m \sum_{j=1}^{D_m} -\frac{1}{2} \mathbf{W}_j^{m\top} (\sum_{i=1}^{N} \mathbf{Z}_{\cdot i} \mathbf{Z}_{\cdot i}^\top) \mathbf{W}_j^m. \tag{3.20}$$

The $\mathbf{W}^m$-independent part of equation (3.18) can be handled as a constant, leaving us with:

$$\frac{\boldsymbol{\tau}_m}{2} \sum_{i=1}^{N} (\mathbf{X}_{\cdot i}^{m\top} \mathbf{W}^m \mathbf{Z}_{\cdot i} + (\mathbf{W}^m \mathbf{Z}_{\cdot i})^\top \mathbf{X}_{\cdot i}^m)$$

$$= \frac{\boldsymbol{\tau}_m}{2} \sum_{i=1}^{N} \sum_{j=1}^{D_m} (\mathbf{X}_{ji}^m \mathbf{W}_j^{m\top} \mathbf{Z}_{\cdot i} + (\mathbf{W}_j^{m\top} \mathbf{Z}_{\cdot i})^\top \mathbf{X}_{ji}^m) \tag{3.21}$$

$$= \boldsymbol{\tau}_m \sum_{j=1}^{D_m} \mathbf{W}_j^{m\top} (\sum_{i=1}^{N} \mathbf{X}_{ji}^m \mathbf{Z}_{\cdot i}). \tag{3.22}$$

Furthermore, the second term of equation (3.18) is

$$-\frac{1}{2}\sum_{k=1}^{K}\boldsymbol{\alpha}_{mk}\mathbf{W}_{\cdot k}^{m\top}\mathbf{W}_{\cdot k}^{m}=-\frac{1}{2}\sum_{j=1}^{D_m}\mathbf{W}_{j}^{m\top}\overline{\overline{\boldsymbol{\alpha}}}_{m}\mathbf{W}_{j}^{m}, \tag{3.23}$$

where $\overline{\overline{\boldsymbol{\alpha}}}_m$ is the mth row of $\boldsymbol{\alpha}$ transferred into a diagonal $K \times K$ matrix.

Thus we get:

$$\begin{aligned}\langle\mathcal{L}_m(\mathbf{W}^m)\rangle_{q(\boldsymbol{\theta})} =&\sum_{j=1}^{D_m}\left[\mathbf{W}_{j}^{m\top}\langle\boldsymbol{\tau}_m\rangle\left(\sum_{i=1}^{N}\mathbf{X}_{ji}^{m}\langle\mathbf{Z}_{\cdot i}\rangle\right)\right.\\ &\left.-\frac{1}{2}\mathbf{W}_{j}^{m\top}\left(\langle\boldsymbol{\tau}_m\rangle\sum_{i=1}^{N}\langle\mathbf{Z}_{\cdot i}\mathbf{Z}_{\cdot i}^{\top}\rangle+\langle\overline{\overline{\boldsymbol{\alpha}}}_m\rangle\right)\mathbf{W}_{j}^{m}\right],\end{aligned} \tag{3.24}$$

from which we can infer that $q(\mathbf{W})$ is of the form:

$$q(\mathbf{W})=\prod_{m=1}^{M}\prod_{j=1}^{D_m}q(\mathbf{W}_{j}^{m})=\prod_{m=1}^{M}\prod_{j=1}^{D_m}\mathcal{N}(\mathbf{W}_{j}^{m}|\mathbf{M}_{mj}^{W},\boldsymbol{\Sigma}_{m}^{W}). \tag{3.25}$$

We get the following update equations for variational distribution $q(\mathbf{W}_j^m)$:

$$\boldsymbol{\Sigma}_{m}^{W}=\left(\langle\boldsymbol{\tau}_m\rangle\sum_{i=1}^{N}\langle\mathbf{Z}_{\cdot i}\mathbf{Z}_{\cdot i}^{\top}\rangle+\langle\overline{\overline{\boldsymbol{\alpha}}}_m\rangle\right)^{-1} \tag{3.26}$$

$$\mathbf{M}_{mj}^{W}=\boldsymbol{\Sigma}_{m}^{W}\langle\boldsymbol{\tau}_m\rangle\left(\sum_{i=1}^{N}\mathbf{X}_{ji}^{m}\langle\mathbf{Z}_{\cdot i}\rangle\right). \tag{3.27}$$

3.1.3 The $q(\boldsymbol{\tau}_m)$ distribution

The complete log-likelihood as a function of $\boldsymbol{\tau}_m$ is

$$\begin{aligned}\mathcal{L}_m(\boldsymbol{\tau}_m)=&\frac{D_m N}{2}\ln\boldsymbol{\tau}_m-\frac{\boldsymbol{\tau}_m}{2}\sum_{i=1}^{N}(\mathbf{X}_{\cdot i}^{m}-\mathbf{W}^{m}\mathbf{Z}_{\cdot i})^{\top}(\mathbf{X}_{\cdot i}^{m}-\mathbf{W}^{m}\mathbf{Z}_{\cdot i})\\ &+(a^{\tau}-1)\ln\boldsymbol{\tau}_m-b^{\tau}\boldsymbol{\tau}_m,\end{aligned} \tag{3.28}$$

with expectation:

$$\begin{aligned}&\langle\mathcal{L}_m(\boldsymbol{\tau}_m)\rangle_{q(\boldsymbol{\theta})}\\ =&(a^{\tau}+\frac{D_m N}{2}-1)\ln\boldsymbol{\tau}_m-\left(b^{\tau}+\frac{1}{2}\sum_{i=1}^{N}\langle(\mathbf{X}_{\cdot i}^{m}-\mathbf{W}^{m}\mathbf{Z}_{\cdot i})^{2}\rangle\right)\boldsymbol{\tau}_m.\end{aligned} \tag{3.29}$$

Thus $q(\boldsymbol{\tau})$ is distributed as

$$q(\boldsymbol{\tau}) = \prod_{m=1}^{M} \mathcal{G}(\boldsymbol{\tau}_m | \mathbf{a}_m^\tau, \mathbf{b}_m^\tau), \tag{3.30}$$

where the parameters are:

$$\mathbf{a}_m^\tau = a^\tau + \frac{D_m N}{2} \tag{3.31}$$

$$\mathbf{b}_m^\tau = b^\tau + \frac{1}{2} \sum_{i=1}^{N} \left\langle (\mathbf{X}_{\cdot i}^m - \mathbf{W}^m \mathbf{Z}_{\cdot i})^2 \right\rangle, \tag{3.32}$$

and the expectation can be computed as

$$\begin{aligned} &\left\langle (\mathbf{X}_{\cdot i}^m - \mathbf{W}^m \mathbf{Z}_{\cdot i})^2 \right\rangle \\ =&\sum_{j=1}^{D_m} \left(\mathbf{X}_{ji}^{m2} - 2\mathbf{X}_{ji}^m \langle \mathbf{W}_j^m \rangle^\top \langle \mathbf{Z}_{\cdot i} \rangle + \operatorname{tr}\left[\langle \mathbf{W}_j^m \mathbf{W}_j^{m\top} \rangle \langle \mathbf{Z}_{\cdot i} \mathbf{Z}_{\cdot i}^\top \rangle \right] \right). \end{aligned} \tag{3.33}$$

3.1.4 The $q(\boldsymbol{\alpha}_m)$ distribution

The $\boldsymbol{\alpha}_m$ dependent parts of the log-likelihood for view m (3.10) are:

$$\begin{aligned} &\frac{D_m}{2} \sum_{k=1}^{K} \ln \boldsymbol{\alpha}_{mk} - \frac{1}{2} \sum_{k=1}^{K} \mathbf{W}_{\cdot k}^{m\top} \boldsymbol{\alpha}_{mk} \mathbf{W}_{\cdot k}^m + \sum_{k=1}^{K} (a^\alpha - 1) \ln \boldsymbol{\alpha}_{mk} - \sum_{k=1}^{K} b^\alpha \boldsymbol{\alpha}_{mk} \\ =&\sum_{k=1}^{K} \left[\frac{D_m}{2} \ln \boldsymbol{\alpha}_{mk} - \frac{1}{2} \mathbf{W}_{\cdot k}^{m\top} \mathbf{W}_{\cdot k}^m \boldsymbol{\alpha}_{mk} + (a^\alpha - 1) \ln \boldsymbol{\alpha}_{mk} - b^\alpha \boldsymbol{\alpha}_{mk} \right] \qquad (3.34) \\ =&\sum_{k=1}^{K} \left(a^\alpha + \frac{D_m}{2} - 1 \right) \ln \boldsymbol{\alpha}_{mk} - \sum_{k=1}^{K} \left(b^\alpha + \frac{1}{2} \mathbf{W}_{\cdot k}^{m\top} \mathbf{W}_{\cdot k}^m \right) \boldsymbol{\alpha}_{mk}. \qquad (3.35) \end{aligned}$$

We get:

$$\left\langle \mathcal{L}(\boldsymbol{\alpha})_{q(\boldsymbol{\theta})} \right\rangle = \sum_{k=1}^{K} \left[\left(a^\alpha + \frac{D_m}{2} - 1 \right) \ln \boldsymbol{\alpha}_{mk} - \left(b^\alpha + \frac{\langle \mathbf{W}_{\cdot k}^{m\top} \mathbf{W}_{\cdot k}^m \rangle}{2} \right) \boldsymbol{\alpha}_{mk} \right], \tag{3.36}$$

from which we can infer that $q(\boldsymbol{\alpha})$ is of the form

$$q(\boldsymbol{\alpha}) = \prod_{m=1}^{M} \prod_{k=1}^{K} q(\boldsymbol{\alpha}_{mk}) = \prod_{m=1}^{M} \prod_{k=1}^{K} \mathcal{G}(\boldsymbol{\alpha}_{mk} | \mathbf{a}_m^\alpha, \mathbf{B}_{mk}^\alpha), \tag{3.37}$$

where

$$\mathbf{a}_m^\alpha = a^\alpha + \frac{D_m}{2} \tag{3.38}$$

$$\mathbf{B}_{mk}^\alpha = b^\alpha + \frac{\langle \mathbf{W}_{\cdot k}^{m\top} \mathbf{W}_{\cdot k}^m \rangle}{2}. \tag{3.39}$$

We can compute the expectation $\langle \mathbf{W}_{\cdot k}^{m\top} \mathbf{W}_{\cdot k}^m \rangle$ by

$$\langle \mathbf{W}_{\cdot k}^{m\top} \mathbf{W}_{\cdot k}^m \rangle = \sum_{j=1}^{p} \left(\mathbf{\Sigma}_m^W + \mathbf{M}_{mj}^W \mathbf{M}_W^{(m,j)\top} \right)_{(k,k)}. \tag{3.40}$$

3.1.5 Convergence of the lower bound

Convergence of the lower bound can be monitored by computing it after each iteration:

$$\mathcal{L}(q) = \int q(\boldsymbol{\theta}) \ln \frac{p(\mathbf{X}, \boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\theta \tag{3.41}$$

$$= \int q(\boldsymbol{\theta}) \ln \frac{p(\mathbf{X}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\theta \tag{3.42}$$

$$= \int \prod_i q(\boldsymbol{\theta}_i) \ln \frac{p(\mathbf{X}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{\prod_i q(\boldsymbol{\theta}_i)} d\theta \tag{3.43}$$

$$= \langle \ln p(\mathbf{X}|\boldsymbol{\theta}) \rangle_{q(\boldsymbol{\theta})} - \sum_i \langle D_{KL}(q(\boldsymbol{\theta}_i) || p(\boldsymbol{\theta}_i)) \rangle_{q(\boldsymbol{\theta}_{-i})}. \tag{3.44}$$

The explicit form of equation (3.44) can be found in Appendix A.

3.2 Correlated-group factor analysis

In group factor analysis the ARD parameters in matrix $\boldsymbol{\alpha}$ of size $M \times K$ are estimated independently for each view-component pair. Given a large number of views in the data, it might be more reasonable to explicitly model their correlations. The key idea behind correlated-group factor analysis is that two views describing the same samples should have something in common. Since their feature spaces differ, we assume that similar views might share similar component activity structure. In this section this novel model is implemented by generating the ARD matrix from a multivariate normal distribution with covariance matrix included in the model. The following

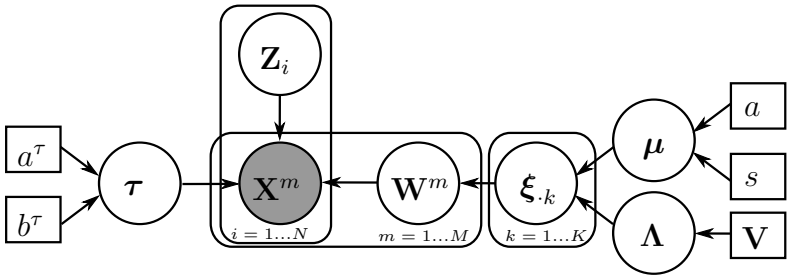


Figure 3.4: Plate model of correlated-group factor analysis (CGFA).

priors are used to replace $\boldsymbol{\alpha}$ in standard group factor analysis:

$$p(\mathbf{W}|\boldsymbol{\xi}_m) \sim \prod_{m=1}^{M}\prod_{k=1}^{K}\mathcal{N}(\mathbf{W}^m_{\cdot k}|\mathbf{0}, \frac{1}{\exp(\boldsymbol{\xi}_{mk})}\mathbf{I}_{D_m}) \tag{3.45}$$

$$p(\boldsymbol{\xi}_{\cdot k}|\boldsymbol{\mu}, \boldsymbol{\Lambda}) \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}^{-1}) \tag{3.46}$$

$$p(\boldsymbol{\mu}|a) \sim \prod_{k=1}^{K}\mathcal{N}(\boldsymbol{\mu}_k|a, s^2) \tag{3.47}$$

$$p(\boldsymbol{\Lambda}|\mathbf{V}) \sim \mathcal{W}(\mathbf{V}, v). \tag{3.48}$$

The model is visualized in figure 3.4. Blei and Lafferty [6] proposed a similar high-level correlation modeling for different topics of a topic model.

For complete model we get the likelihood:

$$p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})/p(\mathbf{X}) \tag{3.49}$$

$$=p(\mathbf{X}|\mathbf{W}, \mathbf{Z}, \boldsymbol{\tau})p(\boldsymbol{\tau}|a^\tau, b^\tau)p(\mathbf{W}|\boldsymbol{\xi})p(\boldsymbol{\xi}|\boldsymbol{\mu}, \boldsymbol{\Lambda})p(\mathbf{Z})p(\boldsymbol{\Lambda}|\mathbf{V})/p(\mathbf{X}) \tag{3.50}$$

$$\begin{aligned}
&= \prod_{m=1}^{M}\left[\prod_{i=1}^{N}\left(|\boldsymbol{\tau}_m\mathbf{I}_{D_m}|^{\frac{1}{2}}e^{-\frac{\boldsymbol{\tau}_m}{2}(\mathbf{X}^m_{\cdot i}-\mathbf{W}^m\mathbf{Z}_{\cdot i})^\top(\mathbf{X}^m_{\cdot i}-\mathbf{W}^m\mathbf{Z}_{\cdot i})}\right)\boldsymbol{\tau}_m^{a^\tau-1}e^{-b^\tau\boldsymbol{\tau}_m}\right.\\
&\left.\prod_{k=1}^{K}\left(e^{\boldsymbol{\xi}_{mk}\frac{D_m}{2}}e^{-\frac{1}{2}\exp(\boldsymbol{\xi}_{mk})\mathbf{W}^{m\top}_{\cdot k}\mathbf{W}^m_{\cdot k}}\right)\right]\prod_{k=1}^{K}\left[|\boldsymbol{\Lambda}|^{\frac{1}{2}}e^{-\frac{1}{2}(\boldsymbol{\xi}_{\cdot k}-\boldsymbol{\mu})^\top\boldsymbol{\Lambda}(\boldsymbol{\xi}_{\cdot k}-\boldsymbol{\mu})}\frac{1}{s}e^{-\frac{(\boldsymbol{\mu}_k-a)^2}{2s^2}}\right]\\
&\prod_{i=1}^{N}e^{-\frac{1}{2}\mathbf{Z}^\top_{\cdot i}\mathbf{Z}_{\cdot i}}|\boldsymbol{\Lambda}|^{\frac{v-M-1}{2}}e^{-\frac{1}{2}\mathrm{tr}(\mathbf{V}^{-1}\boldsymbol{\Lambda})} \times \text{const},
\end{aligned} \tag{3.51}$$

which will be used view-specifically. Thus complete log-likelihood for view

m is:

$$\begin{aligned}\mathcal{L}_m(\mathbf{X},\boldsymbol{\theta}) =& \frac{N}{2}\ln\boldsymbol{\tau}_m - \frac{\boldsymbol{\tau}_m}{2}\sum_{i=1}^{N}(\mathbf{X}^m_{\cdot i} - \mathbf{W}^m\mathbf{Z}_{\cdot i})^\top(\mathbf{X}^m_{\cdot i} - \mathbf{W}^m\mathbf{Z}_{\cdot i}) \\ &+ \frac{D_m}{2}\sum_{k=1}^{K}\boldsymbol{\xi}_{mk} - \frac{1}{2}\sum_{k=1}^{K}\exp(\boldsymbol{\xi}_{mk})\mathbf{W}^{m\top}_{\cdot k}\mathbf{W}^m_{\cdot k} \\ &+ (a^\tau - 1)\ln\boldsymbol{\tau}_m - b^\tau\boldsymbol{\tau}_m + \text{const.}\end{aligned} \tag{3.52}$$

The complete log-likelihood is a sum of (3.10) over views $m = 1...M$ plus a view-independent expression

$$\begin{aligned}\mathcal{L}(\mathbf{Z}) =& -\frac{1}{2}\sum_{i=1}^{N}\mathbf{Z}^\top_{\cdot i}\mathbf{Z}_{\cdot i} + \frac{K}{2}\ln|\boldsymbol{\Lambda}| - \frac{1}{2}\sum_{k=1}^{K}(\boldsymbol{\xi}_{\cdot k} - \boldsymbol{\mu})^\top\boldsymbol{\Lambda}(\boldsymbol{\xi}_{\cdot k} - \boldsymbol{\mu}) \\ &+ \frac{v - M - 1}{2}\ln|\boldsymbol{\Lambda}| - \frac{1}{2}\text{tr}(\mathbf{V}^{-1}\boldsymbol{\Lambda}) - \frac{K}{2}\ln s^2 - \sum_{k=1}^{K}\frac{(\boldsymbol{\mu}_k - a)^2}{2s^2}.\end{aligned} \tag{3.53}$$

Inference for the variational distributions of $\mathbf{Z}$, $\boldsymbol{\tau}$ and $\mathbf{W}$ remains the same as in standard group factor analysis, with the one exception of $\boldsymbol{\alpha}_{mk}$ being replaced by $e^{\boldsymbol{\xi}_{mk}}$. Update equations for the other parameters are derived in the following sections.

3.2.1 The $q(\boldsymbol{\xi})$ distribution

The parts of the complete log-likelihood dependent of $\boldsymbol{\xi}_{\cdot k}$ are:

$$\sum_{m=1}^{M}\left[\frac{D_m}{2}\boldsymbol{\xi}_{mk} - \frac{1}{2}\exp(\boldsymbol{\xi}_{mk})\mathbf{W}^{m\top}_{\cdot k}\mathbf{W}^m_{\cdot k}\right] - \frac{1}{2}(\boldsymbol{\xi}_{\cdot k} - \boldsymbol{\mu}_k)^\top\boldsymbol{\Lambda}(\boldsymbol{\xi}_{\cdot k} - \boldsymbol{\mu}_k). \tag{3.54}$$

Equation (3.54) is not the logarithm of any standard probability distribution, so we will approximate the $q(\boldsymbol{\xi})$ distribution with another variational distribution, analogously to what Blei and Lafferty [6] proposed to a topic model:

$$q(\boldsymbol{\xi}) = \prod_{m=1}^{M}\prod_{k=1}^{K}q(\boldsymbol{\xi}_{mk}) = \prod_{m=1}^{M}\prod_{k=1}^{K}\mathcal{N}(\boldsymbol{\xi}_{mk}|\boldsymbol{\lambda}_{mk}, \mathbf{S}^2_{mk}). \tag{3.55}$$

Since we cannot obtain a closed form optimum given the other parameters any more, we need to maximize the lower bound as presented in equation

(2.11). That is the expectation of (3.54), given $q(\boldsymbol{\xi})$, plus the entropy of $q(\boldsymbol{\xi})$, i.e. $\frac{1}{2}\sum_{m=1}^{M}\sum_{k=1}^{K}\ln \mathbf{S}^2_{mk}$. For $\boldsymbol{\xi}_{\cdot k}$, this becomes:

$$\sum_{m=1}^{M}\left[\frac{D_m}{2}\boldsymbol{\lambda}_{mk} - \frac{1}{2}e^{\boldsymbol{\lambda}_{mk}+\frac{\mathbf{s}^2_{mk}}{2}}\langle \mathbf{W}^{m\top}_{\cdot k}\mathbf{W}^{m}_{\cdot k}\rangle\right] + \frac{1}{2}\sum_{m=1}^{M}\ln \mathbf{S}^2_{mk}$$
$$-\frac{1}{2}\langle \boldsymbol{\xi}^{\top}_{\cdot k}\boldsymbol{\Lambda}\boldsymbol{\xi}_{\cdot k} - \boldsymbol{\mu}^{\top}_{k}\boldsymbol{\Lambda}\boldsymbol{\xi}_{\cdot k} - \boldsymbol{\xi}^{\top}_{\cdot k}\boldsymbol{\Lambda}\boldsymbol{\mu}_k + \boldsymbol{\mu}^{\top}_{k}\boldsymbol{\Lambda}\boldsymbol{\mu}_k\rangle, \tag{3.56}$$

where the last term equals

$$-\frac{1}{2}\boldsymbol{\lambda}^{\top}_{\cdot k}\langle\Lambda\rangle\boldsymbol{\lambda}_{\cdot k} - \frac{1}{2}\mathrm{diag}(\boldsymbol{\Lambda})^{\top}\mathbf{S}^2_{\cdot k} + \boldsymbol{\lambda}^{\top}_{\cdot k}\langle\Lambda\rangle\boldsymbol{\mu}_k. \tag{3.57}$$

Now we can maximize (3.56) as a function of $[\boldsymbol{\lambda}_{\cdot k}, \mathbf{S}^2_{\cdot k}]$ using L-BFGS-B optimization (bounded $\mathbf{S}^2_{\cdot k} > 0$) with gradients

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}_{\cdot k}} = \frac{D}{2} - \frac{1}{2}e^{\boldsymbol{\lambda}_{\cdot k}+\frac{\mathbf{s}^2_{\cdot k}}{2}}\langle \mathrm{tr}(\mathbf{W}^{\cdot}_{\cdot k}\mathbf{W}^{\cdot\top}_{\cdot k})\rangle - \frac{1}{2}\boldsymbol{\lambda}_{\cdot k}\langle\Lambda\rangle + \langle\Lambda\rangle\boldsymbol{\mu}_k \tag{3.58}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{S}^2_{\cdot k}} = -\frac{1}{4}e^{\boldsymbol{\lambda}_{\cdot k}+\frac{\mathbf{s}^2_{\cdot k}}{2}}\langle \mathrm{tr}(\mathbf{W}^{\cdot}_{\cdot k}\mathbf{W}^{\cdot\top}_{\cdot k})\rangle + \frac{1}{2v_k^2} - \frac{1}{2}\mathrm{diag}(\boldsymbol{\Lambda}), \tag{3.59}$$

where $\mathbf{W}^{\cdot}_{\cdot k}$ denotes a matrix with M rows and $D = \sum_{m=1}^{M} D_m$ columns. The matrix contains the projection weights of component k for each feature in data $\mathbf{X}$.

3.2.2 The $q(\boldsymbol{\mu})$ distribution

The complete log-likelihood as a function of $\boldsymbol{\mu}$ is

$$\mathcal{L}(\boldsymbol{\mu}) = \sum_{k=1}^{K}\left[-\frac{1}{2}(\boldsymbol{\xi}_{\cdot k} - \overline{\boldsymbol{\mu}}_k)^{\top}\boldsymbol{\Lambda}(\boldsymbol{\xi}_{\cdot k} - \overline{\boldsymbol{\mu}}_k) - \frac{(\boldsymbol{\mu}_k - a)^2}{2s^2}\right], \tag{3.60}$$

where $\overline{\boldsymbol{\mu}}_k$ denotes a length M vector, where all the elements are $\boldsymbol{\mu}_k$'s.

The expectation of equation (3.60) given other model parameters is

$$\sum_{k=1}^{K}\left[\left(\sum \boldsymbol{\lambda}^{\top}_{\cdot k}\langle\boldsymbol{\Lambda}\rangle + \frac{a}{s^2}\right)\boldsymbol{\mu}_k - \frac{1}{2}\left(\sum_{i,j=1}^{M}\langle\boldsymbol{\Lambda}_{ij}\rangle + \frac{1}{s^2}\right)\boldsymbol{\mu}_k^2\right]. \tag{3.61}$$

We can see that $q(\boldsymbol{\mu})$ can be updated as:

$$q(\boldsymbol{\mu}) = \prod_{k=1}^{K} q(\boldsymbol{\mu}_k) = \prod_{k=1}^{K}\mathcal{N}(\boldsymbol{\mu}_k|\mathbf{m}_k, t^2), \tag{3.62}$$

where

$$t^2 = \left(\sum_{i,j=1}^{M} \langle \mathbf{\Lambda}_{ij} \rangle + \frac{1}{s^2} \right)^{-1} \tag{3.63}$$

$$\mathbf{a}_k = t^2 \left(\sum \boldsymbol{\lambda}_{\cdot k}^\top \langle \mathbf{\Lambda} \rangle + \frac{a}{s^2} \right). \tag{3.64}$$

3.2.3 The $q(\mathbf{\Lambda})$ distribution

The complete log-likelihood as a function of $\mathbf{\Lambda}$ is

$$\begin{aligned} &\frac{K}{2} \ln |\mathbf{\Lambda}| - \frac{1}{2} \sum_{k=1}^{K} (\boldsymbol{\xi}_{\cdot k} - \overline{\boldsymbol{\mu}}_k)^\top \mathbf{\Lambda} (\boldsymbol{\xi}_{\cdot k} - \overline{\boldsymbol{\mu}}_k) + \frac{v - M - 1}{2} \ln |\mathbf{\Lambda}| - \frac{1}{2} \mathrm{tr}[\mathbf{V}^{-1} \mathbf{\Lambda}] \\ =&\frac{K + v - M - 1}{2} \ln |\mathbf{\Lambda}| - \frac{1}{2} \mathrm{tr} \left[\left(\mathbf{V}^{-1} + \sum_{k=1}^{K} (\boldsymbol{\xi}_{\cdot k} - \overline{\boldsymbol{\mu}}_k)(\boldsymbol{\xi}_{\cdot k} - \overline{\boldsymbol{\mu}}_k)^\top \right) \mathbf{\Lambda} \right]. \end{aligned} \tag{3.65}$$

Thus, $q(\mathbf{\Lambda})$ should be updated as

$$q(\mathbf{\Lambda}) = \mathcal{W}(\mathbf{\Lambda} | \mathbf{F}, n), \tag{3.66}$$

where:

$$\mathbf{F} = \left(\mathbf{V}^{-1} + \sum_{k=1}^{K} \left\langle (\boldsymbol{\xi}_{\cdot k} - \overline{\boldsymbol{\mu}}_k)(\boldsymbol{\xi}_{\cdot k} - \overline{\boldsymbol{\mu}}_k)^\top \right\rangle \right)^{-1} \tag{3.67}$$

$$n = K + v, \tag{3.68}$$

and

$$\left\langle (\boldsymbol{\xi}_{\cdot k} - \overline{\boldsymbol{\mu}}_k)(\boldsymbol{\xi}_{\cdot k} - \overline{\boldsymbol{\mu}}_k)^\top \right\rangle = \mathrm{diag}(\mathbf{S}_{\cdot k}^2) + \boldsymbol{\lambda}_{\cdot k} \boldsymbol{\lambda}_{\cdot k}^\top - \boldsymbol{\lambda}_{\cdot k} \overline{\boldsymbol{\mu}}_k^\top - \overline{\boldsymbol{\mu}}_k \boldsymbol{\lambda}_{\cdot k}^\top + \overline{\boldsymbol{\mu}}_k \overline{\boldsymbol{\mu}}_k^\top. \tag{3.69}$$

3.2.4 Lower bound

The lower bound is computed so that we can monitor its convergence. The derivations for this can be found in Appendix B.

3.3 Low-rank relevance determination

The motivation of correlated-group factor analysis was that the $M \times K$ ARD matrix should not be modeled independently, since there will probably be

some dependencies between different views for large M. However, CGFA has two major drawbacks:

- There are approximately $\frac{M^2}{2}$ free parameters to be estimated in $\mathbf{\Lambda}$.
- There is no suitable value for $\mathbf{\Lambda}$'s prior scale matrix $\mathbf{V}$.

The first drawback makes the model hard to infer for large M, which was originally the setting that motivated CGFA. A key feature of GFA in general is that there are separate ARD parameters for each view and component. Thus the prior expectation of $\mathbf{\Lambda}, v\mathbf{V}$, should contain small values corresponding to large variances. Since $v > M - 1$, the prior $\mathbf{V}$ should be set to $\epsilon\mathbf{I}_M$, where ϵ is a small value, to allow this. This, however, would make the prior very informative, as we can see in update equation (3.67).

Since modeling the structure of the ARD matrix via a multivariate normal distribution has major shortcomings, an alternative generative model should be thought of. The motivation for the structured $\boldsymbol{\alpha}$-matrix is still the same: all views of a large collection are probably not independent. Since we assume large M, it would be ideal if the structure could be modeled with $\mathcal{O}(M)$ parameters. This is achieved with one of the main contributions of this thesis: low-rank GFA (LRGFA). The ARD matrix is modeled in LRGFA by fixing it as

$$\boldsymbol{\alpha} = e^{\mathbf{UV}}, \tag{3.70}$$

where the matrix dimensions are $\mathbf{U} \in \mathbb{R}^{M \times R}$ and $\mathbf{V} \in \mathbb{R}^{R \times K}$, $R < M$. The exponentiation is done elementwise, and it is needed to get the scale of $\boldsymbol{\alpha}$ large enough, and to allow only positive values. Instead of MK free parameters (or $\mathcal{O}(\frac{M^2}{2})$ in CGFA) only $MR+KR$ are needed for the ARD. Both $\mathbf{U}$ and $\mathbf{V}$ get a $\mathcal{N}(0, \frac{1}{\lambda})$ prior for their elements, where λ is used to control the variance of $\mathbf{U}$ and $\mathbf{V}$.

The complete log-likelihood with respect to $\boldsymbol{\alpha}$ becomes:

$$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{m=1}^{M} \left[\frac{D_m}{2} \sum_{k=1}^{K} \ln \boldsymbol{\alpha}_{mk} - \frac{1}{2} \sum_{k=1}^{K} \boldsymbol{\alpha}_{mk} \mathbf{W}_{\cdot k}^{m\top} \mathbf{W}_{\cdot k}^{m} \right] \tag{3.71}$$

$$= \sum_{m=1}^{M} \left[\frac{D_m}{2} \sum_{k=1}^{K} (\mathbf{UV})_{mk} - \frac{1}{2} \sum_{k=1}^{K} e^{(\mathbf{UV})_{mk}} \mathbf{W}_{\cdot k}^{m\top} \mathbf{W}_{\cdot k}^{m} \right]. \tag{3.72}$$

The complete log-likelihood with respect to $\mathbf{U}$ and $\mathbf{V}$ equals (3.72), plus the log-priors

$$-\sum_{m=1}^{M} \sum_{r=1}^{R} \frac{\lambda}{2} \mathbf{U}_{mr}^2 - \sum_{r=1}^{R} \sum_{k=1}^{K} \frac{\lambda}{2} \mathbf{V}_{rk}^2 = -\frac{\lambda}{2} \left(\mathrm{tr}(\mathbf{U}^\top \mathbf{U}) + \mathrm{tr}(\mathbf{V}^\top \mathbf{V}) \right). \tag{3.73}$$

In vanilla GFA the ARD parameter $\boldsymbol{\alpha}$ is chosen to be gamma-distributed, since it is the conjugate prior distribution for the precision of normal distribution. Now this is not the case, and the variational inference has to be done in the following manner: First all model parameters except $\boldsymbol{\alpha}$ are updated as in standard GFA, given $\boldsymbol{\alpha}$. Then $\boldsymbol{\alpha}$ is updated by maximizing the expected complete log-likelihood:

$$\boldsymbol{\alpha} = \arg\max_{\boldsymbol{\alpha}} \langle \ln p(\mathbf{X}, \boldsymbol{\theta} | \boldsymbol{\alpha}) \rangle_q, \tag{3.74}$$

which corresponds to maximizing (2.11) when the variational distribution is a point estimate. This type of inference was discussed by Archambeau and Bach [2], and equation (3.74) can be optimized with a numerical optimization method, such as L-BFGS-B [8].

An alternative way for doing variational inference with exponentiated variables was presented by Dikmen and Févotte, who derived closed form update equations for maximizing a lower bound of $\mathcal{L}$ [12]. This type of solution would reduce the computational time of the implementation, since numerical optimization is not needed. However, since it requires an approximation of the lower bound, it has to be considered only if computational time becomes a problem in the low-rank GFA.

3.3.1 Lower bound

The lower bound of the low-rank model can be computed as in GFA, with the following changes: $\boldsymbol{\alpha}$ is replaced with $e^{\mathbf{UV}}$ in $p(\mathbf{W}|\boldsymbol{\alpha})$, there is no $p(\boldsymbol{\alpha})$, and finally for $\mathbf{U}$ we get:

$$D_{KL}(q(\mathbf{U})||p(\mathbf{U})) = \int q(\mathbf{U}) \ln p(\mathbf{U}) dU - \int q(\mathbf{U}) \ln q(\mathbf{U}) d\mathbf{U} = \ln p(\mathbf{U}), \tag{3.75}$$

since $q(\mathbf{U})$ can be thought of as Dirac delta at point $\mathbf{U}$. The divergence between the prior and the variational distribution of $\mathbf{V}$ is likewise $\ln p(\mathbf{V})$, and hence the log-prior term in equation 3.73 will be a part of the lower bound.

3.4 Summary

We introduced Group Factor Analysis and presented two novel extensions for it. Both the extensions are motivated mainly by a scenario where the data consists of many views (large M), and thus it is usually not restrictive to

assume that there are some dependent views. The relation of two views cannot be modeled via their projection matrices, since the views have different feature spaces. The approach in this thesis is to model the ARD matrix such that it has dependency structure with respect to views. In correlated-group factor analysis this is done in a straightforward manner by sampling $\boldsymbol{\alpha}_{\cdot k}$ from a multivariate normal distribution for each $k = 1...K$. This approach is not ideal, since modeling the precision matrix of the normal distribution requires approximately $\frac{M^2}{2}$ free parameters. Additionally an uninformative prior for the precision matrix would lead to small variance in $\boldsymbol{\alpha}$, and thus in a failed automatic relevance determination. Low-rank GFA models $\boldsymbol{\alpha}$ as a product of two rank R matrices, using only $\mathcal{O}(M)$ free parameters. Thus it should be suitable for modeling data where the views are correlated and induce a low-rank component activity structure. The extensions are compared in an artificial data experiment in chapter 5.

Chapter 4

Transfer Learning with GFA

In this chapter GFA is extended to transfer learning setups, where the main goal is to extract relevant information from background data to help model the data of interest. This is done first by standard sequential Bayesian learning and then by introducing a more high-level transfer learning scheme.

4.1 Sequential Bayesian learning

In Bayesian statistics the posterior of model parameters $\boldsymbol{\theta}$ can be learned at once given data $\mathbf{x}$, or sequentially given data $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$, which are independent given $\boldsymbol{\theta}$. Sequential learning is based on repeatedly applying Bayes' rule: first we get a posterior for the parameters given just $\mathbf{x}_1$, then this is used as a prior to get a posterior given $\mathbf{x}_2$ [14]. After n repetitions we get

$$p(\boldsymbol{\theta}|\mathbf{x}_n...\mathbf{x}_1) = \frac{p(\mathbf{x}_n|\boldsymbol{\theta})...p(\mathbf{x}_1|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{x}_n)...p(\mathbf{x}_1)} = \frac{p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{x})}. \tag{4.1}$$

As can be seen in equation (4.1), the posterior of the model parameters can be learned equivalently either by using all the data at once or sequentially. This is, of course, given that we can solve the exact posterior. The sequential approach is needed for example when the data are not available all at once.

In our experiments we are interested in a scenario where there is a background data that has some identical samples or features with the data of interest. This kind of sharing is visualized in terms of the factor model structure in figure 4.1. If all the samples or all the views are shared, the background data could of course be concatenated with the target data but in case of large size the computational time will be long. In a sequential setup the posterior of the shared background model parameters is used as a prior for the target data.

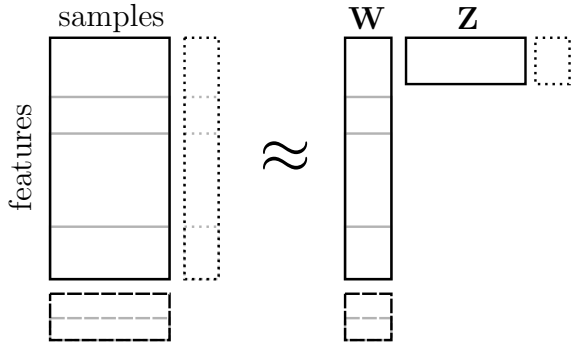


Figure 4.1: Boxes with solid strokes describe the generative model of target data, namely $\mathbf{X} \approx \mathbf{WZ}$. Boxes with dotted and dashed lines describe the background data with shared features and samples, respectively. Gray lines are used to denote the grouping of features into views.

In terms of model parameters we have two cases. Common samples have the same latent representation, and thus the background data can be taken into account by setting a prior for $\mathbf{Z}$ when modeling the target data. Common features or views share the same projection matrix, allowing us to set a prior for $\mathbf{W}$:

$$p(\mathbf{Z}|\mathcal{M}) \sim \prod_{i=1}^{N} \mathcal{N}(\mathbf{Z}_{\cdot i}|\hat{\mathbf{M}}_i^Z, \hat{\mathbf{\Sigma}}^Z) \tag{4.2}$$

$$p(\mathbf{W}|\boldsymbol{\alpha}, \mathcal{M}) \sim \prod_{m=1}^{M} \prod_{k=1}^{K} \mathcal{N}(\mathbf{W}_{\cdot k}^m|\hat{\mathbf{M}}_{m\cdot k}^W, \frac{1}{\boldsymbol{\alpha}_{mk}} \hat{\mathbf{\Sigma}}_m^W), \tag{4.3}$$

where $\mathcal{M}$ denotes a GFA model of the background data. The model contains all the parameters of the variational distributions. $\hat{\mathbf{\Sigma}}^Z$ is a block diagonal matrix with two parts: posterior of the variance of $q(\mathbf{Z})$ from background model and identity matrix for new samples. Since the variational distribution of $\mathbf{W}$ is factorized with respect to features instead of components, $\hat{\mathbf{\Sigma}}_m^W$ will be a diagonal matrix with average prior variances for all the D_m features. New features will have a value of 1. For simplicity, we will denote the inverse matrices of $\hat{\mathbf{\Sigma}}^Z$ and $\hat{\mathbf{\Sigma}}_m^W$ by $\hat{\mathbf{\Lambda}}^W$ and $\hat{\mathbf{\Lambda}}^Z$, respectively. Only the inverse matrices will be used from now on.

With these priors we get the following complete log-likelihood for view

m:

$$\begin{aligned}\mathcal{L}_m(\mathbf{X}, \boldsymbol{\theta}) =& \frac{N}{2} \ln \boldsymbol{\tau}_m - \frac{\boldsymbol{\tau}_m}{2} \sum_{i=1}^{N} (\mathbf{X}_{\cdot i}^m - \mathbf{W}^m \mathbf{Z}_{\cdot i})^\top (\mathbf{X}_{\cdot i}^m - \mathbf{W}^m \mathbf{Z}_{\cdot i}) \\ & - \frac{1}{2} \sum_{k=1}^{K} \boldsymbol{\alpha}_{mk} (\mathbf{W}_{\cdot k}^m - \hat{\mathbf{M}}_{m\cdot k}^W)^\top \hat{\boldsymbol{\Lambda}}_m^W (\mathbf{W}_{\cdot k}^m - \hat{\mathbf{M}}_{m\cdot k}^W) \\ & + \frac{D_m}{2} \sum_{k=1}^{K} \ln(\boldsymbol{\alpha}_{mk} \hat{\boldsymbol{\Lambda}}_m^W) + \sum_{k=1}^{K} (a^\alpha - 1) \ln \boldsymbol{\alpha}_{mk} - \sum_{k=1}^{K} b^\alpha \boldsymbol{\alpha}_{mk} \\ & + (a^\tau - 1) \ln \boldsymbol{\tau}_m - b^\tau \boldsymbol{\tau}_m + \text{const.} \end{aligned} \tag{4.4}$$

The complete log-likelihood for the whole model is a sum of (4.4) over views $m = 1...M$ plus a view-independent term

$$\mathcal{L}(\mathbf{Z}) = -\frac{1}{2} \sum_{i=1}^{N} (\mathbf{Z}_{\cdot i} - \hat{\mathbf{M}}_i^Z)^\top \hat{\boldsymbol{\Lambda}}^Z (\mathbf{Z}_{\cdot i} - \hat{\mathbf{M}}_i^Z). \tag{4.5}$$

The variational Bayesian update equations for sequential Bayesian GFA are derived in the following sections. Similar to the previous chapter, we approximate the posterior using the mean-field assumption that all the parameters are independent. No further assumptions are made regarding the variational distribution q, but it will be further factorized due to the factorized complete likelihood. The priors in sequential Bayesian GFA change the update equations of $\mathbf{Z}$, $\mathbf{W}$ and $\boldsymbol{\alpha}$.

4.1.1 The $q(\mathbf{Z})$ distribution

Variational distribution $q(\mathbf{Z})$ is of the form

$$q(\mathbf{Z}) = \prod_{i=1}^{N} q(\mathbf{Z}_{\cdot i}) = \prod_{i=1}^{N} \mathcal{N}(\mathbf{Z}_{\cdot i} | \mathbf{M}_i^Z, \boldsymbol{\Sigma}^Z), \tag{4.6}$$

and the update equations are:

$$\boldsymbol{\Sigma}^Z = \left(\hat{\boldsymbol{\Lambda}}^Z + \sum_{m=1}^{M} \langle \boldsymbol{\tau}_m \rangle \langle \mathbf{W}^{m\top} \mathbf{W}^m \rangle \right)^{-1} \tag{4.7}$$

$$\mathbf{M}_i^Z = \sum_{m=1}^{M} \boldsymbol{\Sigma}^Z \left(\langle \mathbf{W}^m \rangle^\top \langle \boldsymbol{\tau}_m \rangle \mathbf{X}_{\cdot i}^m + \hat{\boldsymbol{\Lambda}}^Z \hat{\mathbf{M}}_i^Z \right). \tag{4.8}$$

4.1.2 The $q(\mathbf{W}^m)$ distribution

Distribution $q(\mathbf{W}^m)$ is of the form:

$$q(\mathbf{W}^m) = \prod_{j=1}^{D_m} q(\mathbf{W}_j^m) = \prod_{j=1}^{D_m} \mathcal{N}(\mathbf{W}_j^m | \mathbf{M}_{mj}^W, \mathbf{\Sigma}_m^W). \tag{4.9}$$

We get the following update equations:

$$\mathbf{\Sigma}_m^W = \left(\langle \boldsymbol{\tau}_m \rangle \sum_{i=1}^{N} \langle \mathbf{Z}_{\cdot i} \mathbf{Z}_{\cdot i}^\top \rangle + \langle \overline{\overline{\boldsymbol{\alpha}}}_m \rangle \hat{\mathbf{\Lambda}}_m^W \right)^{-1} \tag{4.10}$$

$$\mathbf{M}_{mj}^W = \mathbf{\Sigma}_m^W \left(\langle \boldsymbol{\tau}_m \rangle \left(\sum_{i=1}^{N} \mathbf{X}_{ji}^m \langle \mathbf{Z}_{\cdot i} \rangle \right) + \langle \overline{\overline{\boldsymbol{\alpha}}}_m \rangle \hat{\mathbf{\Lambda}}_m^W \hat{\mathbf{M}}_{mj}^W \right). \tag{4.11}$$

4.1.3 The $q(\boldsymbol{\alpha}_m)$ distribution

Variational distribution $q(\boldsymbol{\alpha})$ is of the form

$$q(\boldsymbol{\alpha}) = \prod_{m=1}^{M} \prod_{k=1}^{K} q(\boldsymbol{\alpha}_{mk}) = \prod_{m=1}^{M} \prod_{k=1}^{K} \mathcal{G}(\boldsymbol{\alpha}_{mk} | \mathbf{a}_m^\alpha, \mathbf{B}_{mk}^\alpha), \tag{4.12}$$

where

$$\mathbf{a}_k^\alpha = a^\alpha + \frac{D_m}{2} \tag{4.13}$$

$$\mathbf{B}_{mk}^\alpha = b^\alpha + \frac{\langle \mathbf{W}_{\cdot k}^{m\top} \hat{\mathbf{\Lambda}}_m^W \mathbf{W}_{\cdot k}^m \rangle + \hat{\mathbf{M}}_{m\cdot k}^{W\top} \hat{\mathbf{\Lambda}}_m^W \hat{\mathbf{M}}_{m\cdot k}^W}{2} - \langle \mathbf{W}_{\cdot k}^{m\top} \rangle \hat{\mathbf{\Lambda}}_m^W \hat{\mathbf{M}}_{m\cdot k}^W. \tag{4.14}$$

4.2 Transferring view correlation structure

Sequential Bayesian learning can be used when background and target data share some samples or features. The CGFA and low-rank GFA extensions however offer chance for a whole new type of transfer learning setup: transferring knowledge when there is nothing directly shared between the domains but the views still have something in common. In our biological experiment this is motivated by a new measurement platform where the views correspond to the same pathways, but have a different feature space. Some of the genes are shared, so sequential learning would be feasible by placing a prior for

the shared parts of $\mathbf{W}$. However, this way we would ignore the knowledge that all the new features in one view are still measuring the same underlying process.

Our approach in this section is to ignore the feature representation of the views and use higher level information to model the view relations. Using CGFA this can be done by using the posterior of $\mathbf{\Lambda}$ from the background model as the prior of the target model (model parameter $\mathbf{V}$). This is sensible since we have assumed that the views are paired between the different domains and measure the same underlying process. More generally, manner the prior $\hat{\mathbf{\Lambda}}$ can be acquired from any GFA model by calculating the inverse sample correlation matrix of $\boldsymbol{\alpha}$. This corresponds to our idea of treating two views similar if they have a similar component activity structure.

In low-rank GFA the component activities of the views are modeled with matrix $\mathbf{U} \in \mathbb{R}^{M \times R}$, and thus the view correlation prior can be taken into account by setting

$$p(\mathbf{U}) \sim \prod_{r=1}^{R} \mathcal{N}(\mathbf{U}_{\cdot r} | \mathbf{0}, \frac{1}{\lambda} \hat{\mathbf{\Lambda}}^{-1}), \tag{4.15}$$

where the strength of the prior can be controlled via the precision parameter λ.

For both LRGFA and CGFA transferring the view correlation structure rewards the model for having similar structure in the target data. In CGFA this kind of transfer of knowledge is straightforward: if two views have a correlation -0.8 between their component activities, we would prefer to have a similar correlation in the target model too. The motivation behind low-rank GFA was that we need $\mathcal{O}(M^2)$ parameters for modeling view correlation this way. The smaller number of parameters, $MR+KR$, affects the nature of this transfer learning setup: by setting a prior we no longer wish that two correlated views have similar component activity, but instead a similar presentation in a low dimensional space spanning the component activities. This works ideally in cases where the views have only few different component activity profiles, allowing LRGFA to to use one column of $\mathbf{U}$ for each profile. It should be noted, though, that $\mathbf{U}$ is normally distributed and thus the component activities of one view are formed as a linear combination of R different profiles, where the rth profile is $\mathbf{V}_r$. Thus LRGFA should work for non-trivial data even with a small R.

Transferring view correlation structure can be done in CGFA via parameter transfer, which is a well-established transfer learning approach [29]. A similar kind of an approach has been implemented for example for learning multiple tasks at once [13]. The knowledge transfer in LRGFA can be viewed as deep learning, since instead of setting a prior for a parameter we set one

for the structure of a parameter [3]. In a related deep learning publication background data was used for learning structural regularities in the form of Markov logic [9].

Chapter 5

Experiments on Artificial Data

Artificial data experiments are used to test whether GFA is able to model data with known parameters and properties as expected. Thus artificial data is generated directly from the model, allowing us to test how good the implementations are when the distributional assumptions are correct.

Lower bound is used in all the GFA models for monitoring the convergence of the variational inference. It can also be used to select the GFA model that is the closest to the actual posterior. However, the lower bounds have no meaningful scale when comparing two models with different priors, for example GFA and low-rank GFA. A very general way of comparing models is via predictive error, which will be used in this thesis the following way:

1. Generate data $\mathbf{X}_{train}$ and $\mathbf{X}_{test}$, with N and 100 samples, respectively.
2. Run GFA given data $\mathbf{X}_{train}$, get model $\mathcal{M}$.
3. Compute $p(\mathbf{Z}_{test}|\mathbf{X}_{test}^{-m}, \mathcal{M})$, where view m is left out of $\mathbf{X}_{test}$.
4. Compute $\mathbb{E}(\mathbf{X}_{test}^{m}|\mathbf{Z}_{test}, \mathcal{M})$.
5. Return the RMSE.

This is repeated for all the views (M times). The predictive error is an ideal measure since it encourages optimal fit, punishing over- and underfitting [31].

Based on the update equations of the latent components in equation (3.17), we can see that the expected mean for $\mathbf{Z}_{(test)i}$ is:

$$\mathbb{E}(\mathbf{Z}_{(test)\cdot i}) = \sum_{n \neq m} \mathbf{\Sigma}^{Z} \langle \mathbf{W}^{n} \rangle^{\top} \langle \boldsymbol{\tau}_{n} \rangle \mathbf{X}^{n}_{(test)\cdot i}, \tag{5.1}$$

where view m is left out since we have assumed centered data: the expectation of $\mathbf{X}^m_{(test)\cdot i}$ is zero. Expectation for the missing view is:

$$\mathbb{E}(\mathbf{X}^m_{test}) = \langle \mathbf{W}^m \rangle \langle \mathbf{Z}_{test} \rangle. \tag{5.2}$$

5.1 High number of views

The CGFA and low-rank models were motivated by the independent ARD parameter inference in standard GFA: when the data consist of a large number of dependent views, modeling the dependencies explicitly could increase model quality. The performances of these models were tested with data consisting of a small amount of samples ($N = 30$) in relatively many views ($M = 40$), with seven features in each view (D_m). The 40 views were divided into four groups with equal size, such that each view in the same group had the same component activity. The component activities were generated in matrix $\boldsymbol{\alpha}$ as four different binary vectors, corresponding to the view groups. In the data generation $\boldsymbol{\alpha}$ is used as the component variance matrix, thus value 0 corresponds to having an infinite ARD parameter. Matrix $\boldsymbol{\alpha}$ is visualized in figure 5.1(a). The other parameters were generated as:

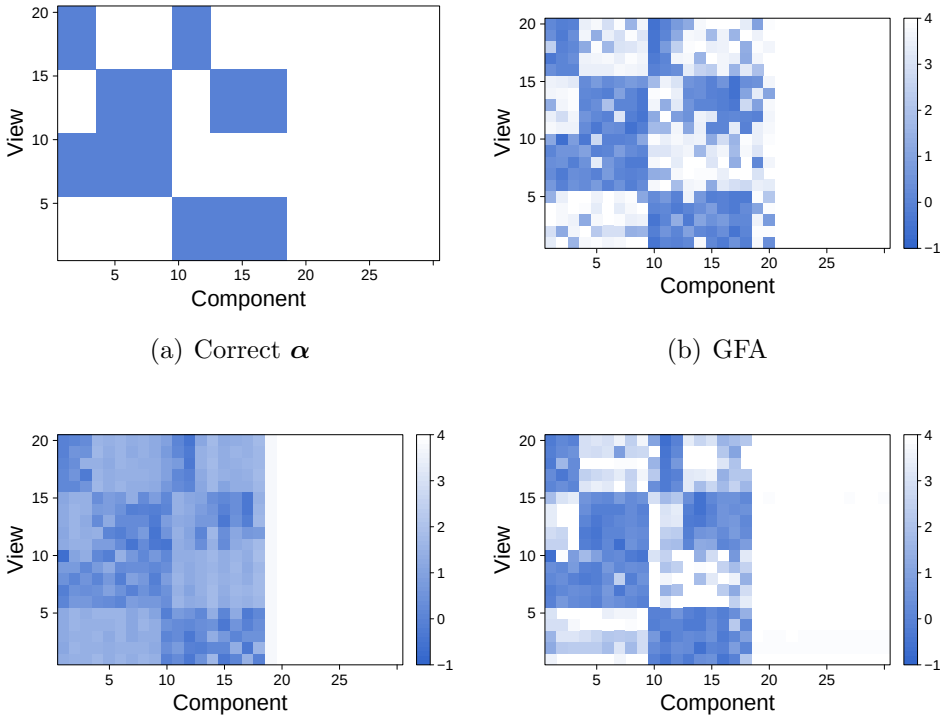
$$\mathbf{Z} \sim \prod_{i=1}^{N} \mathcal{N}(\mathbf{Z}_{\cdot i} | \mathbf{0}, \mathbf{I}_K) \tag{5.3}$$

$$\mathbf{W} \sim \prod_{m=1}^{M} \prod_{k=1}^{K} \mathcal{N}(\mathbf{W}^m_{\cdot k} | \mathbf{0}, \boldsymbol{\alpha}_{mk} \mathbf{I}_{D_m}) \tag{5.4}$$

$$\mathbf{X} \sim \prod_{m=1}^{M} \prod_{i=1}^{N} \mathcal{N}(\mathbf{X}^m_{\cdot i} | \mathbf{W}^m \mathbf{Z}_{\cdot i}, \mathbf{I}_{D_m}). \tag{5.5}$$

The generated data was modeled with GFA, CGFA and low-rank GFA using different numbers of views, ranging from 4 to 40. The predictive RMSE values as a function of the number of views are plotted in figure 5.2. In order to see how well the models estimate the correct parameters, the expected ARD matrices are visualized in figure 5.1 along with the correct matrix.

As expected, low-rank GFA achieved a lower predictive error that standard GFA when the number of views was high. Correlated-group factor analysis, however, performed worse than GFA, even though the data should be optimal for it. This is probably due to the two drawbacks discussed earlier: poor prior for $\boldsymbol{\Lambda}$ and the need to estimate $\mathcal{O}(M^2)$ free parameters for the Wishart-distribution. It is worth noting that all the models work well if there is enough data. However, since the drawbacks of CGFA make the



(a) Correct $\boldsymbol{\alpha}$

(b) GFA

(c) CGFA

(d) LRGFA

Figure 5.1: The ARD matrix $\boldsymbol{\alpha}$ used in data generation is shown in (a); blue corresponds to active components and white inactive. The expectations $\log_{10}\langle\boldsymbol{\alpha}\rangle$ acquired from the GFA models are in (b)-(d). Strong activities are blue and practically inactive components are white. All the models find the correct amount of components. CGFA cannot set components that are active to some views to be fully inactive for others. The ARD structure of LRGFA is closest to the correct $\boldsymbol{\alpha}$.

model lack in performance, it will not be used for the rest of this thesis. After all, low-rank GFA can model correlated views too, and it produced the best predictions in this test. These results are supported by the estimated component activities: GFA models the structure rather correctly, LRGFA even better, but CGFA the worst.

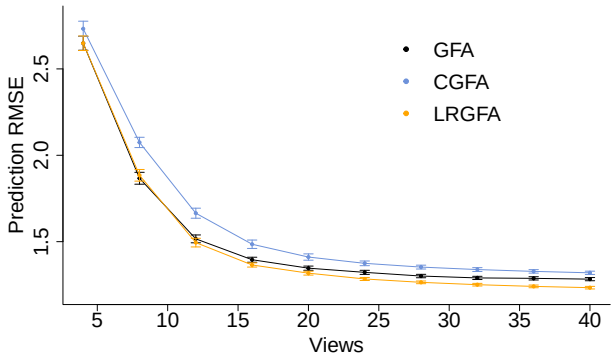


Figure 5.2: Prediction RMSE versus number of views for GFA, CGFA and low-rank GFA. Confidence intervals of the means are based on 50 different artificial data sets. Low-rank GFA is consistently the best model; for example with 40 views its average RMSE is over six standard deviations better than that of GFA.

5.2 Sequential learning

Suppose we have two sets of data coming from the same distribution: background data with N samples and target with 30. If we could solve the exact model posteriors, it would not matter whether the sets were learned at once or sequentially. But as we can only approximate the posterior, we will get the best model for the target by concatenating the sets and modeling them as one. A larger amount of data will help to minimize the error in the posterior approximation. However, if there is an existing model of the background data, we can set a prior for the target data and obtain a good solution quickly. In this section the prediction performance was tested with respect to the amount of background samples for the three models discussed: GFA for target data, GFA for pooled data and sequential Bayesian GFA.

In this experiment the artificial data was generated in a similar manner to section 5.1, with the following data dimensionality: $N = 30, M = 4, D_m = 20$ and $K = 15$. GFA was run for the target data only and for the combined domains. These models were compared to sequential learning with GFA, where model posterior for the background data has been computed beforehand. The predictive RMSE and computational time are compared with respect to the number of background samples N in figure 5.3.

Figure 5.3 shows that sequential GFA performs comparably to GFA applied to concatenated data, given sufficiently large amount of background samples. It is important to note that the online computational time of se-

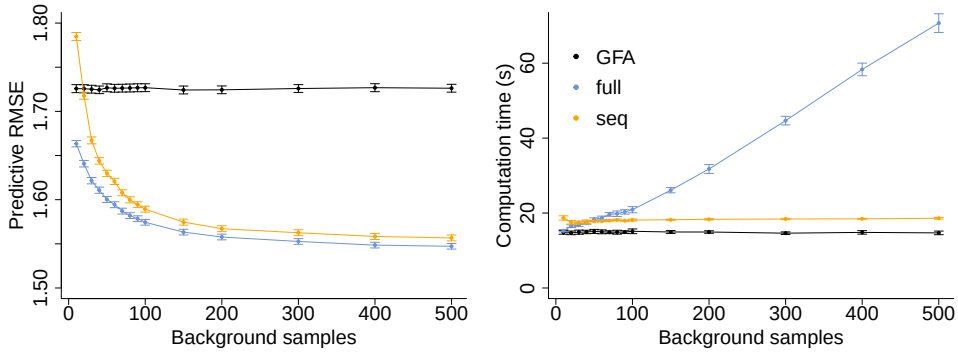


Figure 5.3: Prediction RMSE and computational time versus the number of background samples for three setups: GFA for target data, GFA for concatenated data (*full*) and sequential learning with GFA (*seq*). Confidence intervals of the means are based on 20 different artificial data. GFA for the combined domains is consistently the best model, but once there are enough background samples sequential learning has comparable accuracy. Modeling the combined domains gets considerably slower when there is much more background than target data.

quential GFA depends of the target data only. Thus, given a huge background model, using sequential Bayesian learning is advisable.

5.3 Background data with different features

In the previous section we assumed that background and target data come from the same distribution, which will make combining them the optimal solution. However, a more general framework can be defined by generating target projections matrices as:

$$\mathbf{W}_{\text{target}} = s\mathbf{W}_{\text{background}} + (1-s)\mathbf{W}_{\text{new}}, \tag{5.6}$$

where $s \in [0, 1]$ describes the similarity between target and background. When s is one, the data in this experiment equals the data presented in section 5.1. The goal of this experiment is to find out what can be done when sequential Bayesian learning is no longer applicable. We get the baseline by ignoring the background data completely, but low-rank GFA with a prior Λ can hopefully do better than that.

Figure 5.4 shows that GFA for combined domains works only if the target and background features are very similar. Sequential transfer is more robust,

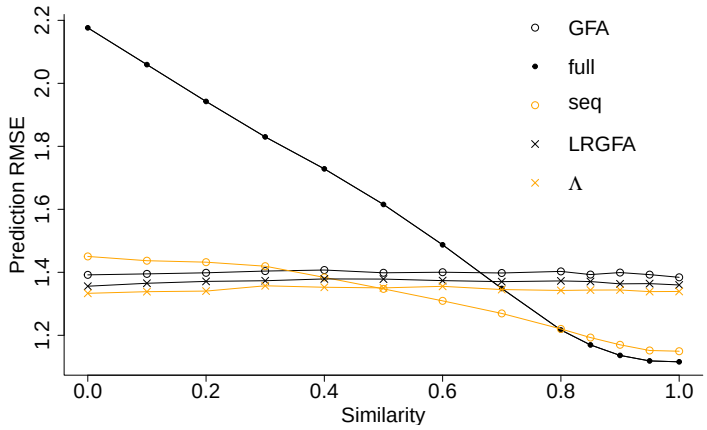


Figure 5.4: Prediction RMSE versus the similarity of target and background. Transfer learning setups are plotted in orange, *seq* denoting sequential Bayesian learning and Λ transferring the view-correlation structure. Full model is optimal when the features in different domains are equivalent. Sequential learning is more robust to dissimilar features.

since even with zero similarity we are only transferring a wrong prior instead of modeling samples with different distributions together. LRGFA models the data better than GFA since the views are grouped. Furthermore, transferring the view-correlation structure is the optimal modeling procedure when the features between the domains are very dissimilar. The prior weight λ was selected from the set $\{0.1, 1, 10\}$ as the one that resulted in the smallest prediction error (λ=1). This does not induce overfitting, since λ was picked from a set of only three values of different scale.

5.4 Summary

We tested the GFA models presented in this thesis on three different types of artificial data. The first test revealed that the implementations of both CGFA and LRGFA are sensible, providing roughly as accurate models as GFA. The artificial data consisted of views divided in four different groups, with up to ten views in a group. This type of setup is ideal for LRGFA, which managed to predict missing views of new samples with the lowest error. The data suits the modeling assumptions of CGFA very well too, since the views inside one group have correlation 1 in their component activities. The generative model of CGFA is not ideal since it had consistently the highest predictive error; the

additional model parameters are more of a burden than help when inferring the model posterior.

The sequential version of GFA was proven to work well, when background and target data share the same features, and there is a decent amount of samples in the background data. Combining the domains instead of learning them sequentially is in this case optimal, since the posterior inference is approximate. However, this has two drawbacks: the computation time depends on the size of the background data, and the features need to be the same in both domains. Sequential learning is much more robust to dissimilar features, as demonstrated in the final experiment. Finally, when the features between the domains are not shared, but the views share the same component activity, low-rank GFA with a transferred view-correlation structure results in the optimal GFA model.

Chapter 6

Drug Response Experiment

In this section we apply the GFA models to the drug response data presented by Lamb et al. [22]. The data contains gene expression measurements acquired via two different microarray platforms having different feature spaces. The dimensionality for both the platforms is around 22000, and the sample sizes are 313 and 682. The features can be divided into 217 pathways. We will consider the smaller data set as target data and try to model it as well as possible. The three relevant models are GFA and low-rank GFA with and without a view-correlation prior from the background platform. The prior is justifiable, since both the platforms measure genes grouped in identical pathways.

The data has been preprocessed as explained by Khan et al. [20], and chemical descriptors of the drugs are added as views for the data. The preprocessing includes normalizing the data, discarding genes with high variance in control measurements and bringing in prior information of biological responses. The final number of unique genes is slightly above 1000 for both background and target data, and there are approximately 700 chemical descriptors. The total dimensionality of both domains is around 4500 due to many genes being present in several views. Both the gene expression and chemical descriptor features are centered and have variance 1.

The model comparison is done in the same way as with artificial data: 260 of the 313 samples are modeled, after which one view of new samples is predicted given the other views. This is repeated for all the views and for ten different divisions of data into training and test samples. The drug response data seems to be very complex, since in all the runs all the components were active, up to 500 components tested. Ideally we would just increase the number of components until some of the components are shut off. In this case this is not possible due to the low number of samples: with 260 components the matrix list $\mathbf{W}$ alone has already as many free parameters as there is data

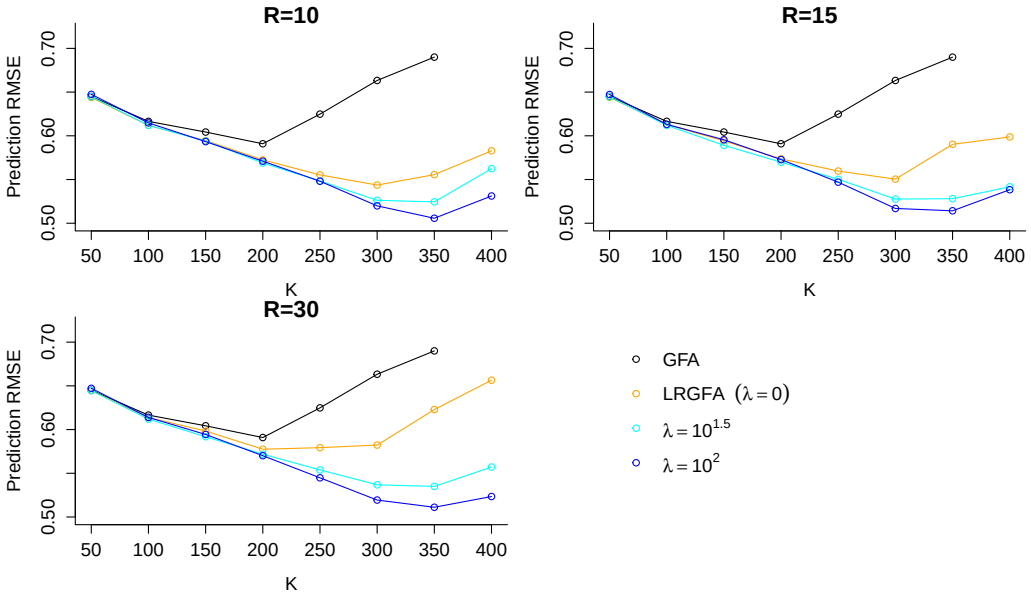


Figure 6.1: Prediction RMSE of GFA models as a function of the number of components. Low-rank GFA was run with three ranks and three prior precisions λ (orange color corresponding to modeling data with no transfer of knowledge). LRGFA is consistently better than standard GFA, and allows modeling the data with more components. Furthermore, transferring the view correlation with precision 100 is consistently the best model, allowing the highest component numbers. It is also more robust to the rank.

available. Thus GFA could overfit the model completely by setting $\mathbf{W}$ as $\mathbf{X}$ and $\mathbf{Z}$ as $\mathbf{I}_K$. This brings up an interesting question concerning factor models: if the purpose of our model is to reduce the data dimensionality, is there any reason to use more parameters than data? From our perspective, the answer is yes. If our data consisted of only two drug response measurements with thousands of genes as features, surely we would not expect that all the genes are connected to only one hidden factor. The problem of having few samples and many features (and latent components) prohibits using standard factor models with large enough component amounts. The three GFA setups were tested to see how well they can model the drug response data with different component amounts. The results are plotted in figure 6.1.

Standard GFA works as expected: as we add more components, we are able to model more complex relations, up until we reach 200 components. Then we experience overfitting, which is expected since the amount of param-

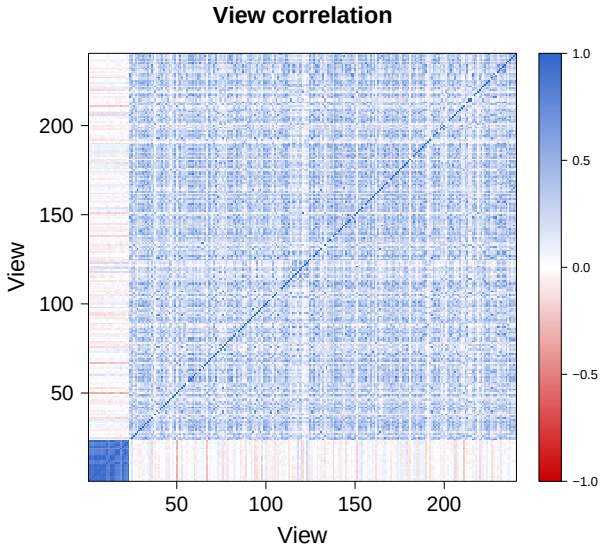


Figure 6.2: View correlation matrix acquired from the background model. The first 24 strongly correlated views are all chemical descriptors of the drugs. All the other views are pathways, mostly independent of the chemical descriptors on the component level.

eters exceeds the amount of data. Low-rank GFA is less prone to overfitting, since it models the component activity matrix with much fewer degrees of freedom. LRGFA modeled the data optimally with around 300 components, resulting in predictive RMSE of approximately 0.55 for ranks 10 and 15. This is significantly lower than 0.60 achieved with GFA using 200 components, as a random guess would have RMSE of 1. Additionally, lower ranks result is less overfitting when more than 300 components are used; values 10 and 15 seem to suit the data well. The view correlation prior further helps to avoid overfitting: a larger amount of components (up to 350) can be used to model the complex data without overfitting. The lowest prediction errors are consistently achieved using the prior with a suitable weight. The lowest RMSE achieved with the GFA models is approximately 0.52. This makes it is interesting to see what kind of information the prior transfers. The view correlation matrix is visualized in figure 6.2.

The view correlation matrix shows a clear structure with respect to the chemical descriptor views: the descriptors share almost exactly the same components, but the correlation between them and the pathways is around zero. It should be noted, though, that zero correlation inferred from the model means that the views have identical component activity for half of

their components. A correlation of -1 is achieved only if the two views have exactly opposite component activity. Thus we can conclude that the chemical descriptors of drugs are relevant while modeling drug response data. Correlations between the pathways are moderate, but some of them were strongly interlinked.

Chapter 7

Discussion

Group factor analysis is an extension of factor analysis to multiple data sets [33]. Instead of modeling dependencies between individual variables, it finds a low-dimensional representation that describes relationships between groups of data sets. Two GFA extensions were introduced in this thesis: correlated-group factor analysis and low-rank GFA. Both the extensions are novel contributions presented in this thesis; they are designed to improve modeling data with a large number of views.

In order to model the relationships between multiple views, GFA needs to learn a matrix $\boldsymbol{\alpha}$ that indicates for each of the K components in which of the M views it is active. Standard GFA infers the component activities of all the views independently, and hence uses MK parameters for describing $\boldsymbol{\alpha}$. For large M this may not be optimal, since often all the views are not independent. The two extensions presented in this thesis are designed to alleviate this problem. Instead of treating each element of $\boldsymbol{\alpha}$ independently, they explicitly model component activity correlations between the views. CGFA does this straightforwardly by sampling component activities from a multivariate distribution with a modeled covariance. In LRGFA the component activity matrix is generated as a product of two low-rank matrices. The design of CGFA suits data with many views poorly, since it uses $\mathcal{O}(M^2)$ parameters for automatic relevance determination. A simple toy data experiment showed that, although of the same order, the predictive performance of CGFA is lacking compared to GFA and its low-rank version. The low-rank extension, however, models correlated views with a small number of parameters ($MR + KR$). The tests showed that it performs statistically significantly better than GFA, given that the data actually has a low-rank component activity matrix.

The other main contribution of this thesis was considering GFA as a transfer learning model. This enables using GFA for knowledge transfer,

when besides target data there is some related background data. Empirical experiments showed that sequential Bayesian learning with GFA has prediction accuracy comparable to GFA learned on both data domains, and computational time comparable to GFA learned on the target data only. Thus sequential learning is recommended when the background data is much larger than the target data. It is also necessary for scenarios where only a subset of the features or samples are shared. Additionally, it was shown that sequential learning is more robust when the distributions of the shared features differ significantly.

Low-rank formulation of GFA allows also another type of transfer learning: transferring the view correlation structure. This is a new GFA extension presented in this thesis. It takes into account more subtle connections between the different data domains: they need to have groups of variables that measure the same view of data, but the actual features need not have a relationship. In our application the need for this type of transfer learning is clear, since the data of two gene expression platforms can be grouped into the same pathways but having different genes. Toy data experiments showed that low-rank GFA with this type of prior is significantly better than without the prior. However, if the features between the domains are the same or very similar, sequential learning and combining the background and target data provide the best results. On the other hand, if the features are dissimilar but the component activities of the views are not, low-rank GFA with view correlation prior is the optimal GFA model.

To demonstrate the novel extensions in practice, we applied them to a drug response experiment, where the target data had only 260 samples but a total dimensionality of approximately 4500. Modeling the high dimensionality would have required an infeasible amount of parameters, since even when the number of parameters was twice the amount of data, none of the components were considered as inactive for all the views. We argue that using more parameters than data may still be reasonable, since a small amount of samples does not imply that the data is simple. However, it makes the inference much more complicated, as was seen in figure 6.1 with standard GFA having increasing prediction error once modeled with enough components. This is expected since the data complexity discourages inactive components, but the small sample size makes the model prone to overfitting. This problem can be dealt with low-rank GFA, since it models the component activity matrix with much fewer degrees of freedom, making it less prone to overfitting. LRGFA achieved consistently lower prediction error than standard GFA; with the most suitable parameter values significantly better. It was also able to model more complex relations in the data (larger K) without losing accuracy.

The drug response data included also background data measured using a different gene expression platform, resulting in different features grouped into identical views (pathways) with the target data. The background model allowed us to transfer the view correlation structure to the target, containing prior information about which pathways and chemical descriptor views should be correlated. The main structure of the view correlation matrix was simple: Chemical descriptor views are heavily correlated with each other, but almost independent of the pathways. The pathways have in general moderate correlations, but some of them were strongly interlinked. This kind of prior information helped further avoiding overfitting. With a suitable prior weight the target data could be modeled using a more complex model than in LRGFA with no loss in predictive accuracy. Using the prior resulted in a smaller prediction error in all the tested cases. Thus LRGFA with a view correlation prior is the optimal GFA model for the drug response data.

In conclusion, GFA is a novel state-of-the-art method for modeling multiple paired data sets. Motivated by the interesting case when there is a high number of data sets, GFA was extended to model the relations between these different data views explicitly in this thesis. Low-rank GFA was extensively validated to perform better than GFA in both artificial and real data experiments. In addition, it reduces the effective number of model parameters, and thus is less prone to overfitting when there is an insufficient amount of samples in the data. We also presented a new transfer learning setup that allows transferring high-level knowledge. When the data domains have only this kind of high-level connection, transferring view correlation prior results in the optimal GFA model.

Bibliography

[1] Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. *Molecular biology of the cell*, 4 ed. Garland Science, 2002.

[2] Archambeau, C., and Bach, F. Sparse probabilistic projections. In *Advances in Neural Information Processing Systems 21* (2009), pp. 73–80.

[3] Arel, I., Rose, D., and Karnowski, T. Deep machine learning - a new frontier in artificial intelligence research. *Computational Intelligence Magazine 5* (2010), 13–18.

[4] Bach, F., and Jordan, M. A probabilistic interpretation of canonical correlation analysis. Tech. rep., Department of Statistics, University of California, Berkeley, 2005.

[5] Bishop, C. *Pattern recognition and machine learning*, vol. 4. Springer, 2006.

[6] Blei, D., and Lafferty, J. Correlated topic models. In *Advances in Neural Information Processing Systems 18* (2006), pp. 147–154.

[7] Boyd, S., and Vandenberghe, L. *Convex optimization*. Cambridge University Press, 2004.

[8] Byrd, R., Lu, P., Nocedal, J., and Zhu, C. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing 16* (1995), 1190–1208.

[9] Davis, J., and Domingos, P. Deep transfer via second-order Markov logic. In *Proceedings of the 26th International Conference on Machine Learning* (2009), pp. 217–224.

[10] Dempster, A., Laird, N., and Rubin, D. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* (1977), 1–38.

[11] Diebold, F. X., and Mariano, R. S. Comparing predictive accuracy. *Journal of Business & Economic Statistics 13* (July 1995), 253–263.

[12] Dikmen, O., and Févotte, C. Nonnegative dictionary learning in the exponential noise model for adaptive music signal representation. In *Advances in Neural Information Processing Systems 24* (2011), pp. 2267–2275.

[13] Evgeniou, A., and Pontil, M. Multi-task feature learning. In *Advances in Neural Information Processing Systems 19* (2007), pp. 41–48.

[14] Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. *Bayesian data analysis*, 2 ed. Chapman and Hall/CRC, 2004.

[15] Guyon, I., and Elisseeff, A. An introduction to variable and feature selection. *The Journal of Machine Learning Research 3* (2003), 1157–1182.

[16] Haghighi, A., Liang, P., Berg-Kirkpatrick, T., and Klein, D. Learning bilingual lexicons from monolingual corpora. In *Association for Computational Linguistics* (2008), pp. 771–779.

[17] Hotelling, H. Relations between two sets of variates. *Biometrika 28* (1936), 321–377.

[18] Jolliffe, I. *Principal component analysis*. Wiley Online Library, 2005.

[19] Kenakin, T. *A pharmacology primer: theory, application and methods*, 2 ed. Academic Press, 2006.

[20] Khan, S., Faisal, A., Mpindi, J., Parkkinen, J., Kalliokoski, T., Poso, A., Kallioniemi, O., Wennerberg, K., and Kaski, S. Comprehensive data-driven analysis of the impact of chemoinformatic structure on the genome-wide biological response profiles of cancer cells to 1159 drugs. *BMC bioinformatics* (2012).

[21] Kullback, S., and Leibler, R. On information and sufficiency. *The Annals of Mathematical Statistics 22* (1951), 79–86.

[22] Lamb, J., Crawford, E., Peck, D., Modell, J., Blat, I., Wrobel, M., Lerner, J., Brunet, J., Subramanian, A., Ross, K., et al. The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease. *Science Signalling 313* (2006), 1929–1935.

[23] Li, T., Zhang, C., and Ogihara, M. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics 20* (2004), 2429–2437.

[24] MacKay, D. Bayesian methods for backpropagation networks. *Models of Neural Networks III 6* (1996), 211–254.

[25] Maldjian, J., Laurienti, P., Kraft, R., Burdette, J., et al. An automated method for neuroanatomic and cytoarchitectonic atlas-based interrogation of fMRI data sets. *Neuroimage 19* (2003), 1233 – 1239.

[26] Molinowski, E. *Factor analysis in chemistry*. Wiley, New York, 1991.

[27] Neal, R. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995.

[28] Nielsen, F. B. Variational approach to factor analysis and related models. Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2004.

[29] Pan, S., and Yang, Q. A survey on transfer learning. *Knowledge and Data Engineering 22* (2010), 1345–1359.

[30] Parisi, G., and Shankar, R. Statistical field theory. *Physics Today 41* (1988).

[31] Sheiner, L., and Beal, S. Some suggestions for measuring predictive performance. *Journal of Pharmacokinetics and Pharmacodynamics 9* (1981), 503–512.

[32] Thurstone, L. Multiple factor analysis. *Psychological Review 38* (1931).

[33] Virtanen, S., Klami, A., Khan, S. A., and Kaski, S. Bayesian group factor analysis. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics* (2012), pp. 1269–1277.

[34] West, M. Bayesian factor regression models in the "large p, small n" paradigm. *Bayesian Statistics 7* (2003), 723–732.

Appendix A

Lower Bound for GFA

The lower bound can be written as:

$$\begin{aligned}\mathcal{L}(q) =&\langle \ln p(\mathbf{X}|\boldsymbol{\theta})\rangle_q - D_{KL}(q(\mathbf{Z})||p(\mathbf{Z})) - \langle D_{KL}(q(\mathbf{W})||p(\mathbf{W}|\boldsymbol{\alpha}))\rangle_{q(\boldsymbol{\alpha})} \\ &- D_{KL}(q(\boldsymbol{\tau})||p(\boldsymbol{\tau}|a^\tau, b^\tau)) - D_{KL}(q(\boldsymbol{\alpha})||p(\boldsymbol{\alpha}|a^\alpha, b^\alpha)),\end{aligned} \tag{A.1}$$

where

$$\langle \ln p(\mathbf{X}|\boldsymbol{\theta})\rangle_q = \sum_{m=1}^{M}\left[-\frac{ND_m}{2}(\ln(2\pi) + \langle \boldsymbol{\tau}_m\rangle) + D_m\left(\langle \boldsymbol{\tau}_m\rangle b_m^\tau - a_m^\tau\right)\right]. \tag{A.2}$$

Using shorthand notation such as $D_{KL}(\mathbf{W}) := \langle D_{KL}(q(\mathbf{W})||p(\mathbf{W}|\boldsymbol{\alpha}))\rangle_{q(\boldsymbol{\alpha})}$, the KL-divergences are:

$$D_{KL}(\mathbf{Z}) = -\frac{N}{2}\ln|\boldsymbol{\Sigma}^Z| - \frac{N}{2}\mathrm{tr}[\mathbf{I}_K - \boldsymbol{\Sigma}^Z] + \frac{1}{2}\sum_{i=1}^{N}\mathrm{tr}[\mathbf{M}_i^Z\mathbf{M}_i^{Z\top}] \tag{A.3}$$

$$\begin{aligned}D_{KL}(\mathbf{W}) = \sum_{m=1}^{M}\Bigg[&-\frac{D_m}{2}\sum_{k=1}^{K}\langle \ln \boldsymbol{\alpha}_{mk}\rangle - \frac{1}{2}\sum_{j=1}^{D_m}\Big(\ln|\boldsymbol{\Sigma}_m^W| \\ &+\mathrm{tr}\left[\mathbf{I}_{D_m} - \left(\boldsymbol{\Sigma}_m^W + \mathbf{M}_{mj}^W\mathbf{M}_{mj}^{W\top}\right)\langle\overline{\overline{\boldsymbol{\alpha}}}_m\rangle\right]\Big)\Bigg]\end{aligned} \tag{A.4}$$

$$\begin{aligned}D_{KL}(\boldsymbol{\tau}) = \sum_{m=1}^{M}&[\ln\Gamma(a^\tau) - a^\tau\ln b^\tau - (a^\tau - 1)\langle \ln \boldsymbol{\tau}_m\rangle + b^\tau\langle \boldsymbol{\tau}_m\rangle \\ &- \ln\Gamma(\mathbf{a}_m^\tau) + \mathbf{a}_m^\tau\ln\mathbf{b}_m^\tau + (\mathbf{a}_m^\tau - 1)\langle \ln \boldsymbol{\tau}_m\rangle - \mathbf{b}_m^\tau\langle \boldsymbol{\tau}_m\rangle]\end{aligned} \tag{A.5}$$

$$\begin{aligned}D_{KL}(\boldsymbol{\alpha}) = \sum_{m=1}^{M}\sum_{k=1}^{K}&[\ln\Gamma(a^\alpha) - a^\alpha\ln b^\alpha - (a^\alpha - 1)\langle \ln \boldsymbol{\alpha}_{mk}\rangle + b^\alpha\langle \boldsymbol{\alpha}_{mk}\rangle \\ &- \ln\Gamma(\mathbf{a}_m^\alpha) + \mathbf{a}_m^\alpha\ln\mathbf{B}_{mk}^\alpha + (\mathbf{a}_m^\alpha - 1)\langle \ln \boldsymbol{\alpha}_{mk}\rangle - \mathbf{B}_{mk}^\alpha\langle \boldsymbol{\alpha}_{mk}\rangle].\end{aligned} \tag{A.6}$$

Appendix B

Lower Bound for CGFA

The lower bound of correlated-group factor analysis has identical data likelihood and KL-divergences for distributions of $\mathbf{Z}, \boldsymbol{\tau}$ and $\mathbf{W}$ as GFA, with the exception of $\boldsymbol{\alpha}$ being replaced with $e^{\boldsymbol{\xi}}$. The other divergences needed to calculate the lower bound are:

$$\begin{aligned}\langle D_{KL}(q(\boldsymbol{\xi})||p(\boldsymbol{\xi}|\boldsymbol{\mu},\boldsymbol{\Lambda}))\rangle_{q(\boldsymbol{\mu})q(\boldsymbol{\Lambda})} =& \frac{1}{2}\sum_{k=1}^{K}\Bigg[\ln\frac{|\boldsymbol{\Lambda}^{-1}|}{|\mathrm{diag}(\mathbf{S}_{\cdot k}^2)|} + \mathrm{tr}[\boldsymbol{\Lambda}\mathrm{diag}(\mathbf{S}_{\cdot k}^2)] - M \\ &+ (\boldsymbol{\lambda}_{\cdot k} - \boldsymbol{\mu}_k)^T\boldsymbol{\Lambda}(\boldsymbol{\lambda}_{\cdot k} - \boldsymbol{\mu}_k)\Bigg] \qquad \text{(B.1)}\\ \langle D_{KL}(q(\boldsymbol{\mu})||p(\boldsymbol{\mu}|a,s))\rangle =& \frac{1}{2}\sum_{k=1}^{K}\left[\ln\frac{s^2}{t^2} + \frac{t^2}{s^2} - 1 + \frac{(\mathbf{m}_k - a)^2}{s^2}\right] \qquad \text{(B.2)}\\ D_{KL}(q(\boldsymbol{\Lambda})||p(\boldsymbol{\Lambda}|\mathbf{V},v)) =& \frac{1}{2}\Big[(n-M-1)L(\mathbf{F},n) + n\mathrm{tr}(\mathbf{V}^{-1}\mathbf{F}) - nM \\ &-(v-M-1)L(\mathbf{V},v) + 2\ln\frac{Z(\mathbf{V},v)}{Z(\mathbf{F},n)}\Big], \qquad \text{(B.3)}\end{aligned}$$

where

$$L(\mathbf{A}, b) = \int \mathrm{Wi}(\mathbf{X}|\mathbf{A}, b) \ln |\mathbf{X}| d\mathbf{X}$$
$$= \sum_{i=1}^{d} \psi(\frac{b+1-i}{2}) + \ln |\mathbf{A}| + d \ln 2 \tag{B.4}$$
$$Z(\mathbf{A}, b) = 2^{bd/2} |\mathbf{A}|^{b/2} \Gamma_d(\frac{b}{2}) \tag{B.5}$$
$$\Gamma_d(\frac{b}{2}) = \pi^{\frac{d(d-1)}{4}} \prod_{j=1}^{d} \Gamma(\frac{b-j+1}{2}). \tag{B.6}$$