Variational Learning for Latent Gaussian Models of Discrete Data

by

Mohammad Emtiyaz Khan

M.Sc., Indian Institute of Science, 2004

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate Studies

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

November 2012

Abstract

This thesis focuses on the variational learning of latent Gaussian models for discrete data. The learning is difficult since the discrete-data likelihood is not conjugate to the Gaussian prior. Existing methods to solve this problem are either inaccurate or slow. We consider a variational approach based on evidence lower bound optimization. We solve the following two main problems of the variational approach: the computational inefficiency associated with the maximization of the lower bound and the intractability of the lower bound. For the first problem, we establish concavity of the lower bound and design fast learning algorithms using concave optimization. For the second problem, we design tractable and accurate lower bounds, some of which have provable error guarantees. We show that these lower bounds not only make accurate variational learning possible, but can also give rise to algorithms with a wide variety of speed-accuracy trade-offs. We compare various lower bounds, both theoretically and experimentally, giving clear design guidelines for variational algorithms. Through application to real-world data, we show that the variational approach can be more accurate and faster than existing methods.

Preface

All the work in this thesis was done under the supervision of Dr. Kevin Murphy. I also collaborated with several other researchers without whose help this work would not have been possible.

- I started working on variational learning in 2009 at Xerox Research Center Europe, under the supervision of Dr. Guillaume Bouchard. In 2010, Dr. Benjamin Marlin and I formulated a more general problem in the context of factor analysis. This resulted in the NIPS publications Khan et al. [2010]. Part of Chapter 6 is based on it.

- The piecewise bounds introduced in Chapter 4 was presented at ICML Marlin et al. [2011]. I am thankful to Ben who originally came up with the idea of piecewise bounds.

- The stick breaking likelihood introduced in Chapter 5 was first mentioned to me by Guillaume Bouchard in 2009. In 2011, Ben convinced me that it could be useful in the context of factor models and Gaussian process regression. Shakir Mohamed helped me with experiments for the paper that was finally published at AISTATS Khan et al. [2012a].

- The fast convergent algorithm of Chapter 3 is published in NIPS Khan et al. [2012b]. I collaborated with Shakir who helped me with the experiments done in the paper.

- I would like to exclusively thank Shakir for useful discussions which resulted in many insights presented in this thesis. Shakir also helped me with many figures and illustrations presented in this thesis.

Table of Contents

List of Tables

List of Figures

List of Algorithms

List of Abbreviations

AIS	Annealed Importance Sampling
CTM	Correlated Topic Model
ELBO	Evidence Lower BOund
EM	Expectation Maximization
EP	Expectation Propagation
GLM	Generalized Linear Model
GP	Gaussian Process
GPC	Gaussian Process Classification
HMC	Hybrid Monte Carlo
IIA	Independence of Irrelevant Alternatives
INLA	Integrated Nested Laplace Approximations
KL	Kullback-Leibler
LDA	Latent Dirichlet Allocation
LGM	Latent Gaussian Models
LGGM	Latent Gaussian Graphical Model
LLP	Logistic-Log-Partition
LSE	Log-Sum-Exp
LVB	Local Variational Bound
MAP	Maximum-a-posteriori
MCEM	Monte Carlo Expectation Maximization
MCMC	Markov Chaing Monte Carlo
MH	Metropolis Hastings
PCA	Principal Component Analysis
POS	Product of Sigmoid
PPCA	Probabilistic Principal Component Analysis
RUM	Random Utility Model
SAEM	Stochastic Approximation Expectation Maximization

Acknowledgements

First of all and most importantly, I would like to thank Kevin Murphy for his supervision and for giving me an opportunity to work with him. I would also like to thank my supervisory committee which includes Nando D. Freitas and Arnaud Doucet for helpful discussions and encouragement.

During my PhD, I have been extremely lucky to collaborate with many amazing people. First of all, I would like to thank Benjamin Marlin for his help and support. Without him, this work would not have been possible. A special thanks goes to Shakir for useful discussions that have resulted in many insights presented in my thesis. I would also like to thank researchers at Xerox Research center Europe at Grenoble, where I first started working on this problem. It was a great experience to work there under the supervision of Guillaume Bouchard who not only helped me understand variational methods, but also introduced me to many new flavors of wine. I would also like to thank Onno Zoeter and Cedric Archambeau for many useful discussions. Many thanks to my colleagues at UBC including David Duvenaud, Frank Hutter, Mark Schmidt, Mike Chiang, Roman Holenstein, and Sancho McCann.

I would like to thank all my friends in Vancouver and back home in India who have helped me throughout the course of my PhD. A special thanks goes to Kath, my coffee buddy, for always being there to cheer me up when I am down and celebrate with me when I am happy.

Finally, I would like to thank my family for their support throughout my life.

Chapter 1

Introduction

The development of accurate models with efficient learning algorithms for high-dimensional and multivariate discrete data is an important and long-standing problem in machine learning and computational statistics. It has applications for data analysis in a wide variety of areas such as econometrics, social science, medical diagnostics, education, multimedia, web, and recommender systems.

In this thesis, we focus on the class of Latent Gaussian Models (LGMs), which model data distributions using Gaussian latent variables. LGMs include popular models such as factor analysis and probabilistic principal components analysis for continuous data [Bartholomew et al., 2011; Tipping and Bishop, 1999], binary and multinomial factor analysis for discrete data [Collins et al., 2002; Khan et al., 2010; Mohamed et al., 2008; Wedel and Kamakura, 2001], and Gaussian process regression and classification [Nickisch and Rasmussen, 2008]. LGMs use a continuous latent space to capture correlation in data, and are therefore well-suited for the analysis of heterogeneous discrete datasets containing different kinds of data such as binary, categorical, ordinal, count, etc. LGMs also allow for a principled approach to handle missing data and can be used for dimensionality reduction, data prediction, and visualization.

The main focus of this thesis is the Bayesian analysis of discrete data using LGMs. A Bayesian analysis is fruitful in applications where uncertainty estimates are important; for example, when the data is noisy or contains missing values. In the Netflix movie-rating dataset, for instance, the number of ratings per user and movie varies significantly: some movies in the training set have as few as 3 ratings, while one user has rated over 17,000 movies[1]. Such variance in missingness makes some of the users (and movies) more informative than others. In situations like this, the Bayesian analysis is useful since it computes a posterior distribution over the unknown variables, providing a measure of reliability and thereby improving prediction performance. Bayesian analysis also computes the marginal likelihood, a ro-

[1] `www.netflixprize.com/community/viewtopic.php?id=141`

bust measure of model-fit, which provides a principled approach for model selection.

In case of LGMs for discrete data, the Bayesian analysis is challenging since the marginalization of latent variables, required for the computation of marginal likelihood, is intractable. This integration can be performed analytically in Gaussian-likelihood LGMs such as factor analysis because the model is jointly Gaussian in the latent and observed variables [Bishop, 2006]. LGMs with discrete-data likelihoods, such as logit and probit, lack this property, resulting in intractable integrals for the marginal likelihood.

There are several approaches for approximating this integral. The most popular approach is to use Markov chain Monte Carlo (MCMC) methods [Albert and Chib, 1993; Frühwirth-Schnatter and Frühwirth, 2010; Holmes and Held, 2006; Mohamed et al., 2008; Scott, 2011]. This approach can be quite accurate since, theoretically, the approximation converges to the truth as the sampler collects more samples. In practice, however, this incurs a significantly high computational cost, and requires expert-level knowledge for the convergence diagnostics and sampler tuning. There exist many deterministic approaches which usually have much lower computational cost than MCMC, but they are less general. For example, the recently proposed Integrated Nested Laplace Approximation (INLA) [Rue et al., 2009], which uses numerical integration to approximate the integral, is limited to six or fewer parameters and is thus not suitable for LGMs with large number of parameters. Similarly, expectation propagation (EP) [Minka, 2001], a method that approximates the posterior distribution by maintaining expectations and iterating until these expectations are consistent for all variables, has problems when applied to the latent factor model case [Rattray et al., 2009]. The EP procedure is not always guaranteed to converge and can be numerically unstable [Jylänki et al., 2011; Seeger and Nickisch, 2011] (see Section 2.3.3 for details).

Our solutions in this thesis are based on the variational approach where we compute a lower bound to the integral using Jensen's inequality. Unlike EP and INLA, this approach does not suffer from numerical issues and convergence problems, and is applicable to more general settings such as parameter learning in the latent factor models. However, similar to other deterministic methods, the computational cost of the variational approach remains much lower than MCMC.

Application of the variational approach to discrete-data LGMs has two main challenges. First, the lower bound obtained using Jensen's inequality may not be tractable and an additional *local variational bound* (LVB) is required for tractability. Although many LVBs have been proposed (for

example, Ahmed and Xing [2007]; Blei and Lafferty [2006]; Bouchard [2007]; Braun and McAuliffe [2010]; Khan et al. [2010]), a clear conclusion on accuracy of these bounds is missing. The variational approach performs poorly whenever LVBs are not accurate, but design of accurate LVBs with error guarantees is a difficult task.

The second challenge with the variational approach is their computational efficiency. Even though the approach is more efficient than MCMC, it still does not scale well to large datasets, limiting its use to datasets with only few thousands of variables. For discrete-data LGMs, a big source of inefficiency comes from the posterior covariances. Variational learning requires inversion of posterior covariances which prohibits the use of the variational approach to large datasets. In addition, these covariances have quadratic number of parameters which slows down the optimization.

In this thesis, we make several contributions to solve the two challenges associated with the variational approach. Our first contribution is the design of accurate LVBs for discrete-data modeling. In Chapter 4,5, and 6, we derive new LVBs for binary, categorical, and ordinal data that lead to accurate and fast variational algorithms. Our second contribution is in the comparison and analysis of LVBs to obtain good design guidelines to construct efficient variational algorithms. We show, through application to the real dataset, that LVBs can be used to obtain variational algorithms with a wide range of speed-accuracy trade-offs. Our third contribution is to improve the computational efficiency of variational algorithms using concave optimization methods. Finally, we thoroughly compare our approach with existing approaches, such as MCMC, EP, and variational Bayes, showing that our approach, while solving the problems associated with existing approaches, performs not only comparably to them, but sometimes even better than them.

Since LGMs are the main focus of this thesis, we spend the rest of the chapter reviewing this model class, along with the statistical and computational challenges it offers. We define LGMs in Section 1.1 and describe a few examples of LGMs in Section 1.2. Our examples include popular models such as Bayesian logistic regression, Gaussian process classification, and principal component analysis. LGMs are used to model discrete data using a likelihood function which "links" the latent Gaussian vector to the discrete data. We discuss different choices of likelihood functions for binary, categorical, ordinal, and count data in Section 1.3. We describe the objectives of the thesis in Section 1.4, listing four main tasks of interest. We conclude the chapter with a brief summary of the thesis in Section 1.5.

1.0.1 Notation

We denote real-valued scalars by small letters e.g. $a, b, c, \ldots$, real-valued vectors by boldface small letters e.g. $\boldsymbol{\lambda}, \boldsymbol{\alpha}, \mathbf{x}, \ldots$, real-valued matrices by boldface capital letter e.g. $\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots$, (subsets of) natural numbers by capital letters e.g. $N, M, \ldots$. We denote i'th element of a vector $\mathbf{a}$ by a_i and (i, j)'th entry of a matrix $\mathbf{A}$ by A_{ij}. Given a square matrix $\mathbf{A}$ of size N and $S \subseteq \{1, 2, \ldots, N\}$, $\mathbf{A}_S$ denotes the matrix formed by taking the rows and columns corresponding to indices in S. Similarly, $\mathbf{A}_{:,S}$ (or $\mathbf{A}_{S,:}$) is a matrix formed by taking columns (or rows) corresponding to indices in S. We denote the determinant of a positive-definite matrix $\mathbf{A}$ by $|\mathbf{A}|$. A random vector $\mathbf{x}$ following a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ is denoted by $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$. A probability distribution over $\mathbf{x}$ given parameters $\boldsymbol{\theta}$ is denoted by $p(\mathbf{x}|\boldsymbol{\theta})$. Expectation of a function $f(\mathbf{x})$ with respect to this distribution is denoted by $\mathbb{E}_{p(x|\theta)}[f(\mathbf{x})]$. Lower and upper bound to a quantity f is denoted by $\underline{f}$ and $\overline{f}$, respectively. We denote the identity matrix of size D by I_D.

1.1 Latent Gaussian Models (LGMs)

In this section, we define the generic latent Gaussian model for discrete data. We consider N data instances, with n'th visible data vector denoted by $\mathbf{y}_n$ and corresponding latent vector denoted by $\mathbf{z}_n$. In general, $\mathbf{y}_n$ and $\mathbf{z}_n$ will have dimensions D and L, respectively. Each element of $\mathbf{y}_n$, denoted by y_{dn}, can take values from a (countably infinite) discrete set S_d, e.g. for binary observations $S_d = \{0, 1\}$ and for count observations $S_d = \{0, 1, 2, \ldots\}$. In LGMs, the latent vectors $\mathbf{z}_n$ follow a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ as shown in Eq. 1.1. The probability of each observation y_{dn} is parameterized in terms of the linear predictor $\boldsymbol{\eta}_{dn}$, as shown in Eqs. 1.2 and 1.3. The predictor $\boldsymbol{\eta}_{dn}$ is defined through $\mathbf{W}_d$, a real-valued matrix of size $K_d \times L$ called the factor loading matrix, and $\mathbf{w}_{0d}$, a K_d length real-valued vector called the offset vector. The value of K_d and the form of the likelihood $p(y_{dn}|\boldsymbol{\eta}_{dn})$ both depend on the type of observation y_{dn}. We give few examples of these in the next section.

$$p(\mathbf{z}_n|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{1.1}$$

$$\boldsymbol{\eta}_{dn} = \mathbf{W}_d \mathbf{z}_n + \mathbf{w}_{0d} \tag{1.2}$$

$$p(\mathbf{y}_n|\mathbf{z}_n, \boldsymbol{\theta}) = \prod_{d=1}^{D} p(y_{dn}|\boldsymbol{\eta}_{dn}, \boldsymbol{\theta}) \tag{1.3}$$

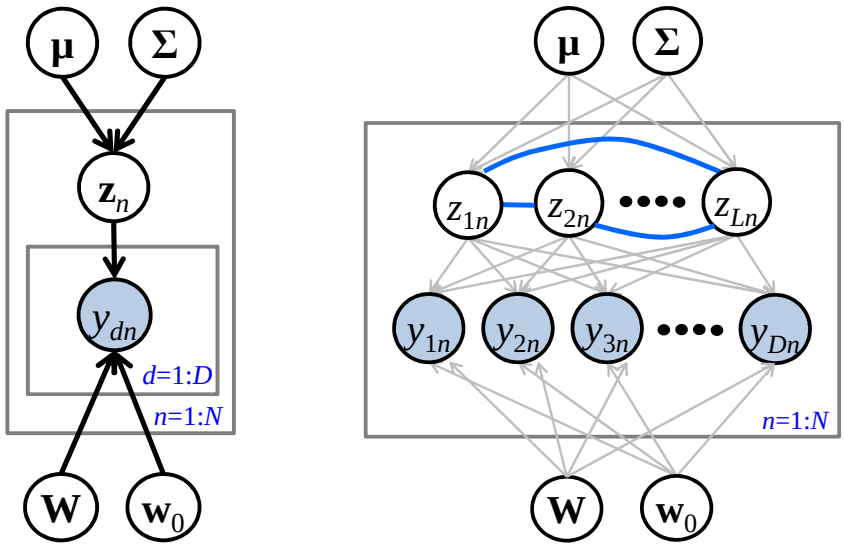


Figure 1.1: The graphical model for latent Gaussian models shown in left figure, and expanded in the right figure to explicitly show the correlation in the latent vector $\mathbf{z}_n$ induced due to a non-diagonal $\mathbf{\Sigma}$.

If extra input features are available, the predictor $\boldsymbol{\eta}_{dn}$ can be redefined as a function of them. For example, in hierarchical regression models such as discrete choice model, features $\mathbf{X}_{dn}$ are available for (d, n)'th data point and the predictor may be defined in terms of these features: $\boldsymbol{\eta}_{dn} = \mathbf{X}_{dn}\mathbf{z}_n$ (see Section 1.2.2). We will stick with our definition, since our definition can be trivially extended to include these special cases (e.g. we can redefine loading matrices as $\mathbf{W}_{dn} = \mathbf{X}_{dn}$).

We denote the set of parameters by $\boldsymbol{\theta}$ which constitutes of parameters required to define the following set of variables: $\{\boldsymbol{\mu}, \mathbf{\Sigma}, \mathbf{W}_d, \mathbf{w}_{0d}\}$ along with some extra parameters needed to define the likelihood of Eq. 1.3.

The graphical model for LGM is shown in Fig. 1.1(a), with an expanded version in Fig. 1.1(b) showing correlations in $\mathbf{z}_n$ due to a non-diagonal $\mathbf{\Sigma}$.

It is redundant to have both $\mathbf{W}$ and a non-diagonal $\mathbf{\Sigma}$ since the data correlation can be modeled with one of them fixed to identity and other being unrestricted [Roweis and Ghahramani, 1999]. However, we allow both $\mathbf{W}$ and $\mathbf{\Sigma}$ to be unrestricted since it allows us to define LGM in the most general setting. Depending on a particular model, we will assume some restriction on these parameters.

1.2 Examples of LGMs

In this section, we give examples of LGMs, establishing the generality of our definition given in the previous section. We discuss many popular models which can be grouped into two major classes: (1) the class of regression models which includes models such as Bayesian logistic regression and Gaussian process classification (2) the class of latent factor models which includes models such as principal component analysis and discrete choice models. The former class is supervised while the latter is unsupervised. Table 1.1 summarizes the equivalence between these models and our generic LGM definition. Our goal in this section is to discuss the challenging problems that these models raise. We summarize these problems in Section 1.4 as our learning objectives for the generic LGM.

1.2.1 Bayesian Logistic Regression

The Bayesian logistic regression model [Holmes and Held, 2006; McCullagh and Nelder, 1989] is used to assign a binary label $y \in \{0, 1\}$ to input variables (or features) $\mathbf{x}$, say of length L. Given D input-output pairs, the distribution of d'th label y_d is defined by the Bernoulli logit likelihood, shown in Eq. 1.4, with the logit function defined in Eq. 1.5. A constant bias term is usually added to $\mathbf{x}_d^T\mathbf{z}$ which we ignore here for simplicity. We assume that $\mathbf{z}$, the weight vector, follows a Gaussian distribution as shown in Eq. 1.6. The hyperparameters of this model are $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$, since we integrate out $\mathbf{z}$.

$$p(y_d = 1|\mathbf{x}_d, \mathbf{z}) = \sigma(\mathbf{z}^T\mathbf{x}_d) \tag{1.4}$$

$$\sigma(\eta) := 1/(1 + \exp(-\eta)) \tag{1.5}$$

$$p(\mathbf{z}|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{1.6}$$

The graphical model is shown in Fig. 1.2(a). See Table 1.1 for equivalence between this model and LGM. The latent vector $\mathbf{z}$ is the regression weight vector, and the observation y_d can be modeled using the predictor $\mathbf{x}_d^T\mathbf{z}$. The Bayesian logistic regression is a special case of LGM with the following restrictions: (1) there are no repeated samples of y_d available i.e. $N = 1$, (2) the loading matrix $\mathbf{w}_d$ is given and is equal to $\mathbf{x}_d^T$, and (3) $\mathbf{w}_0 = \mathbf{0}$. The table also gives the descriptions of various quantities such as N, D, L and K under this model.

This model can be generalized to handle outputs of different types using the generalized linear model (GLM) [McCullagh and Nelder, 1989], all extensions being within the LGM class. For example, for multi-class regression (also known as polychotomous regression in statistics), the multinomial

Model	Data	Latent vector $\mathbf{z}$	$\boldsymbol{\theta}$	N	D	L	K
Bayesian Logistic Regression	$\{y_d, \mathbf{x}_d\}$	Regression weights $y_d \leftarrow f(\mathbf{z}^T \mathbf{x}_d)$	$\boldsymbol{\mu}, \boldsymbol{\Sigma}$	1	#Obs	#Features	#Class
Discrete Choice Model	$\{y_{dn}, \mathbf{X}_{dn}\}$	Regression weights $y_d \leftarrow f(\mathbf{X}_{dn}\mathbf{z}_n)$	$\boldsymbol{\mu}, \boldsymbol{\Sigma}$	#users	#Choices	#Features	#Prod
Gaussian Process Classification	$\{y_d, \mathbf{x}_d\}$	Regression weights $y_d \leftarrow f(z_d)$	s, σ	1	#Obs	#Features	#Class
Probabilistic PCA	$\{y_{dn}\}$	Latent factors $y_{dn} \leftarrow f(\mathbf{W}_d \mathbf{z}_n + \mathbf{w}_{0d})$	$\mathbf{W}, \mathbf{w}_0$	#Obs	#Dims	#Factors	#Cat
Latent Gaussian Graphical Model	$\{y_{dn}\}$	Latent Gaussian $y_{dn} \leftarrow f(z_{dn})$	$\boldsymbol{\mu}, \boldsymbol{\Sigma}$	#Obs	#Dims	#Dims	#Cat
Correlated Topic Model	$\{y_{dn}\}$	Topic vector $y_{dn} \leftarrow f_d(\mathbf{z}_n \vert \mathbf{B})$	$\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{B}$	#Docs	#Words	#Topics	#Vocab

Table 1.1: This table shows many existing models as examples of LGM. Each column is a quantity from our generic LGM definition. Each row shows corresponding quantities for a model. First three models are supervised and last three are unsupervised. For columns 2 and 3, d ranges over 1 to D and n ranges over 1 to N. $\{a_{dn}\}$ denotes the set of variables indexed by all values of d and n. $y \leftarrow f(z)$ implies that y can be generated using some function f of z. In last four columns, 'Obs' means Observations, 'Dims' means Dimensions, 'Docs' means Documents, 'Prod' means Products, 'Cat' means Categories, 'Vocab' means Vocabulary, and '#' represents the number of a quantity.

logit likelihood can be used; see Section 1.3.3 for details on the multinomial logit likelihood.

We now describe various tasks of interests in the Bayesian logistic regression model. First of all, we would like to infer the posterior distribution over $\mathbf{z}$ given the training data, as shown below.

$$p(\mathbf{z}|\mathbf{y}, \mathbf{X}, \boldsymbol{\theta}) \propto p(\mathbf{z}|\boldsymbol{\theta})p(\mathbf{y}|\mathbf{X}, \mathbf{z}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \prod_{d=1}^{D} e^{y_d \mathbf{x}_d^T \mathbf{z} - \log(1+\exp(\mathbf{x}_d^T \mathbf{z}))} \tag{1.7}$$

Here, $\mathbf{y}$ is a vector containing all the output labels and $\mathbf{X}$ is a matrix with corresponding inputs as rows. Since the Gaussian distribution is not conjugate to the Bernoulli logistic likelihood, we do not have a parametric form for the posterior. A popular approach is to use Markov chain Monte Carlo (MCMC) to generate samples from the posterior [Frühwirth-Schnatter and Frühwirth, 2010; Holmes and Held, 2006; Scott, 2011]. An alternative approach is to approximate the posterior distribution using a Gaussian distribution [Jaakkola and Jordan, 1996; Minka, 2001].

Another task of interest is the prediction of new inputs. Given an input vector $\mathbf{x}_*$, the predictive distribution for label y_* can be obtained as in Eq. 1.8. This integral is intractable but can be simplified to a one-dimension integral which can be approximated either by numerical integration or Monte Carlo estimate [Bishop, 2006, Section 4.5.2].

$$p(y_* = 1|\mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}) = \int p(y_* = 1|\mathbf{x}_*, \mathbf{z})p(\mathbf{z}|\mathbf{y}, \mathbf{X}, \boldsymbol{\theta})d\mathbf{z} \tag{1.8}$$

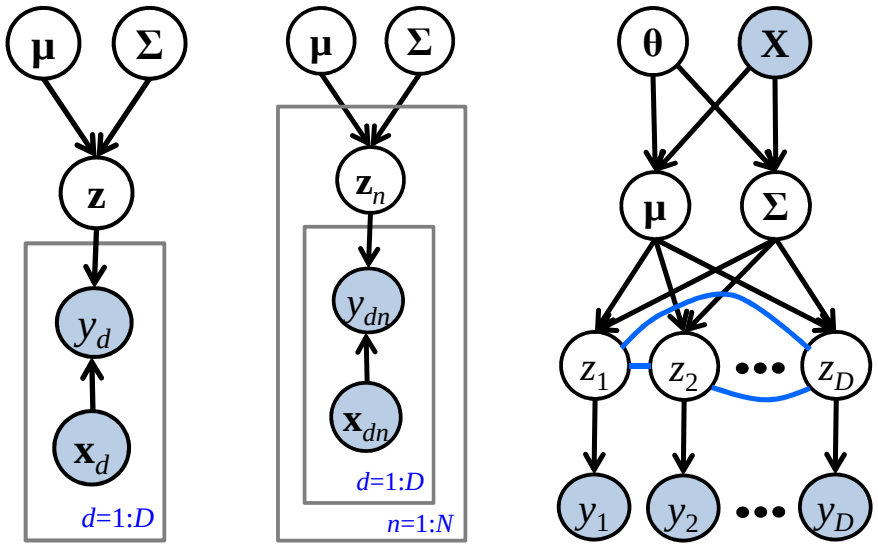
$$= \int \sigma(\mathbf{x}_*^T \mathbf{z})p(\mathbf{z}|\mathbf{y}, \mathbf{X}, \boldsymbol{\theta})d\mathbf{z} \tag{1.9}$$

Finally, we would like to compute marginal likelihood for model comparison and selection [Frühwirth-Schnatter and Wagner, 2008]. This involves a high-dimensional intractable integral shown below.

$$p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = \int p(\mathbf{z}|\boldsymbol{\theta})p(\mathbf{y}|\mathbf{X}, \mathbf{z})d\mathbf{z} \tag{1.10}$$

$$= \int \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \prod_{d=1}^{D} e^{y_d \mathbf{x}_d^T \mathbf{z} - \log(1+\exp(\mathbf{x}_d^T \mathbf{z}))} d\mathbf{z} \tag{1.11}$$

Although this remains a difficult problem to solve, approximations based on MCMC methods can be obtained (see Frühwirth-Schnatter and Wagner [2008] for details). Alternatively, variational methods can be used to obtain an approximation or a lower bound to the integral [Jaakkola and Jordan, 1996; Minka, 2001].



(a) (b) (c)

Figure 1.2: Examples of LGMs for supervised learning: (a) Bayesian logistic regression (b) discrete choice model (c) Gaussian process classification.

1.2.2 Discrete Choice Model

Discrete choice models [Train, 2003] are used in the analysis of consumer choice data which arises when agents select items from a finite collection of alternatives, e.g. people buying items from a store. Heterogeneous discrete choice models that allow preferences to differ across agents are based on a hierarchical regression model and are a special case of LGMs.

We describe the model defined by Braun and McAuliffe [2010]. Let us say that we have agents, indexed by $n = 1, 2, \ldots, N$, choosing items, indexed by $k = 1, 2, \ldots, K$, differentiated according to L features or attributes. We observe the choices that agents make at different times, i.e. we have pairs $(y_{dn}, \mathbf{X}_{dn})$ where y_{dn} is the item chosen by the n'th agent at d'th decision and $\mathbf{X}_{dn}$ is a $K \times L$ matrix containing attributes for all items encountered by the user n when she made her d'th choice. The feature $\mathbf{X}_{dn}$ can also be constant for all d and n, but a more realistic situation is to allow it to be different for each (d, n) pair. For simplicity, let us assume that the total number of decisions for all users is same and equal to D.

Under the "random utility model", the utility of n'th user choosing the k'th item is obtained as follows: $u_{kdn} = \mathbf{z}_n^T \mathbf{x}_{kdn} + e_{kdn}$, where $\mathbf{z}_n$ is L-

length vector of the user specific "taste" or "preference loadings", $\mathbf{x}_{kdn}$ is k'th row of $\mathbf{X}_{dn}$, and e_{kdn} is the random error. A priori, we expect that the taste of agents will be similar and hence we assume a multivariate Gaussian prior over $\mathbf{z}_n$ shown in Eq. 1.12. We assume that random errors e_{kdn} are iid from a Gumbel Type 2 distribution which leads to the multinomial logit distribution for the choices as shown in Eq. 1.13 (see Section 1.3.3 for details on the multinomial logit distribution). The parameter set of this model is given by $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$.

$$p(\mathbf{z}_n|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{1.12}$$

$$p(y_{dn} = k|\mathbf{X}_{dn}, \mathbf{z}_n) = \frac{\exp\left(\mathbf{z}_n^T \mathbf{x}_{kdn}\right)}{\sum_{j=1}^{K} \exp\left(\mathbf{z}_n^T \mathbf{x}_{jdn}\right)} \tag{1.13}$$

The graphical model is shown in Fig. 1.2(b). See Table 1.1 for equivalence between this model and LGM. The latent vector $\mathbf{z}_n$ is the regression weight vector for n'th user, and the observation y_{dn} can be modeled using the predictor $\mathbf{X}_{dn}\mathbf{z}_n$. This model is a special case of LGM with the following modifications: allow $\mathbf{W}_d$ to depend on n, and define $\mathbf{W}_{dn} = \mathbf{X}_{dn}$ and $\mathbf{w}_0 = \mathbf{0}$. The table also gives the descriptions of various quantities such as N, D, L and K under this model.

We now discuss the tasks of interests in this model. For notational convenience, we use a dummy encoding for y_{dn}, that is, we encode it as a binary vector where we set $\widetilde{y}_{kdn}$ to 1, if $y_{dn} = k$, and set rest of the elements of $\widetilde{\mathbf{y}}$ to 0. Using this, the multinomial logit likelihood can be more compactly written as shown below,

$$p(y_{dn}|\boldsymbol{\eta}_{dn}) = \frac{\exp(\tilde{\mathbf{y}}_{dn}^T \boldsymbol{\eta}_{dn})}{\sum_{j=1}^{K} \exp\left(\eta_{jdn}\right)} = \exp\left(\tilde{\mathbf{y}}_{dn}^T \boldsymbol{\eta}_{dn} - \mathrm{lse}(\boldsymbol{\eta}_{dn})\right) \tag{1.14}$$

where $\mathrm{lse}(\boldsymbol{\eta}) := \log \sum_j \exp(\eta_j)$ is the log-sum-exp (LSE) function. Also define $\mathbf{X}_n$ to be the matrix containing the $\mathbf{X}_{dn}$ as rows, $\mathbf{X}$ to be the matrix containing all the $\mathbf{X}_n$, $\mathbf{y}_n$ to be vector of all the $\tilde{\mathbf{y}}_{dn}$, and $\mathbf{y}$ to be the vector of all the $\mathbf{y}_n$.

First task of interest is the computation of posterior distribution of $\mathbf{z}_n$ given $\mathbf{y}_n$, which can be rewritten as follows,

$$p(\mathbf{z}_n|\mathbf{y}_n, \mathbf{X}_n, \boldsymbol{\theta}) \propto p(\mathbf{z}_n|\boldsymbol{\theta}) p(\mathbf{y}_n|\mathbf{X}_n, \mathbf{z}_n) \tag{1.15}$$

$$= \mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \prod_{d=1}^{D} \exp\left(\tilde{\mathbf{y}}_{dn}^T \boldsymbol{\eta}_{dn} - \mathrm{lse}(\boldsymbol{\eta}_{dn})\right) \tag{1.16}$$

Note that, similar to the logistic regression model, we do not have a closed form expression for this posterior distribution, the reason being the same: the Gaussian distribution is not conjugate to the multinomial logit distribution.

Second task of interest is the computation of marginal likelihood of $\mathbf{y}_n$. This involves computation of an intractable integral as shown below.

$$p(\mathbf{y}_n|\boldsymbol{\theta}, \mathbf{X}_n) = \int \mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \prod_{d=1}^{D} \exp\left(\tilde{\mathbf{y}}_{dn}^T \boldsymbol{\eta}_{dn} - \text{lse}(\boldsymbol{\eta}_{dn})\right) d\mathbf{z}_n \tag{1.17}$$

The above marginal likelihood is also useful in the estimation of parameters $\boldsymbol{\theta}$. Fully Bayesian approaches assume a prior distribution over $\boldsymbol{\theta}$ and aim to compute a posterior distribution shown below.

$$p(\boldsymbol{\theta}|\mathbf{y}, \mathbf{X}) \propto p(\boldsymbol{\theta}) \prod_{n=1}^{N} p(\mathbf{y}_n|\boldsymbol{\theta}, \mathbf{X}_n) \tag{1.18}$$

We see that computation of this posterior distribution is difficult since it requires $p(\mathbf{y}_n|\boldsymbol{\theta}, \mathbf{X}_n)$ for all n, each of which involves an intractable integral. We can obtain samples from this distribution using MCMC methods although this usually tends to be slow and does not scale well with N and D. An alternative approach is to compute a point estimate of $\boldsymbol{\theta}$, instead of a full posterior distribution, e.g. the maximum-a-posteriori (MAP) estimate [Braun and McAuliffe, 2010] shown below.

$$\hat{\boldsymbol{\theta}} = \arg\max_{\theta} \log p(\boldsymbol{\theta}) + \sum_{n=1}^{N} \log p(\mathbf{y}_n|\boldsymbol{\theta}, \mathbf{X}_n) \tag{1.19}$$

We can use deterministic methods to obtain an approximation of the marginal likelihood and its gradient with respect to $\boldsymbol{\theta}$, which can then be used in a numerical optimizer to obtained an estimate of $\boldsymbol{\theta}$.

Finally, we are interested in computing predictive choice distribution which is the distribution of an item given a new attribute matrix $\mathbf{X}_*$, as shown in Eq. 1.20. This can also be interpreted as the choices made by an "average agent" [Braun and McAuliffe, 2010].

$$p(y_* = k|\mathbf{X}_*, \boldsymbol{\theta}) = \int p(y_* = k|\mathbf{z}, \boldsymbol{\theta}) p(\mathbf{z}|\mathbf{X}_*, \boldsymbol{\theta}) d\mathbf{z} \tag{1.20}$$

1.2.3 Gaussian Process Classification (GPC)

The Bayesian logistic regression model is a linear classifier since y depends on a linear function of $\mathbf{x}$. A Gaussian process classification model uses a non-linear latent function $f(\mathbf{x})$ to obtain the distribution of y [Kuss and Rasmussen, 2005; Nickisch and Rasmussen, 2008]. The non-linearity in $f(\mathbf{x})$ is obtained by assuming a Gaussian process prior.

A Gaussian process [Rasmussen and Williams, 2006] is a stochastic process fully specified by a mean function $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ and a positive-definite covariance function $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$. Assuming a Gaussian process prior over $f(\mathbf{x})$ implies that a random variable is associated with every input $\mathbf{x}$, such that given all the inputs $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_D]$, the joint distribution over $[f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_D)]$ is Gaussian.

Let $z_d = f(\mathbf{x}_d)$ and $\mathbf{z} = (z_1, z_2, \ldots, z_D)$. The Gaussian process prior is shown in Eq. 1.21. Here, $\boldsymbol{\mu}$ is a vector with $m(\mathbf{x}_i)$ as the i'th element, $\boldsymbol{\Sigma}$ is a matrix with $k(\mathbf{x}_i, \mathbf{x}_j)$ as the (i, j)'th entry, and θ are the parameters of the mean and covariance function (more details given below). As before, the likelihood of a label is obtained using the Bernoulli logit likelihood, as shown in Eq. 1.22.

$$p(\mathbf{z}|\mathbf{X}, \theta) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{1.21}$$

$$p(y_d = 1|z_d) = \sigma(z_d) \tag{1.22}$$

The detailed description of mean and covariance function can be found in Rasmussen and Williams [2006]. An example of the mean function is the zero-mean function with $m(\mathbf{x}) = 0, \forall \mathbf{x}$. Similarly, an example of the covariance function is the squared-exponential covariance function (also known as radial-basis function or Gaussian) defined below,

$$k(\mathbf{x}, \mathbf{x}') = s \exp[-\tfrac{1}{2l}(\mathbf{x} - \mathbf{x}')^T(\mathbf{x} - \mathbf{x}')] \tag{1.23}$$

For this function, the hyper-parameters of the GP prior are $\boldsymbol{\theta} = \{s, l\}$, both positive real numbers, representing the signal variance and length scale, respectively. Similar to the Bayesian logistic regression case, likelihood function can be modified to handle various types of outputs, for example, see Chu and Ghahramani [2005]; Girolami and Rogers [2006].

The graphical model is shown in Fig. 1.2(c) which can be compared against the LGM graphical model in Fig. 1.1(b). See Table 1.1 for equivalence between this model and LGM. The latent vector $\mathbf{z}$ contains the latent variables z_d for observations y_d. The observation y_d is modeled using z_d. There is one latent variable per observations, making the latent dimension

equal to the data dimension. This model is a special case of LGM with the following restrictions: (1) there are no repeated samples of y_d available i.e. $N = 1$, (2) the loading vector $\mathbf{W}_d = \mathbf{I}_D$ and $\mathbf{w}_0 = \mathbf{0}$, and (3) the features $\mathbf{X}$ are used to define $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ using mean and covariance functions, respectively. The table also gives the descriptions of various quantities such as N, D, L and K under this model.

The various tasks of interests are similar to the Bayesian logistic regression model. Let $\mathbf{y}$ be the vector y_d and $\mathbf{X}$ be the matrix containing $\mathbf{X}_d$ as rows. First of all, we are interested in posterior inference which has an intractable form shown below.

$$p(\mathbf{z}|\mathbf{y}, \mathbf{X}, \boldsymbol{\theta}) \propto p(\mathbf{z}|\mathbf{X}, \boldsymbol{\theta})p(\mathbf{y}|\mathbf{z}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \prod_{d=1}^{D} e^{y_d z_d - \log(1+\exp(z_d))} \quad (1.24)$$

Similar to the regression case, we can use MCMC or deterministic approximations. Detailed comparisons of these methods can be found in Kuss and Rasmussen [2005] and Nickisch and Rasmussen [2008]. Note that the dimensionality of $\mathbf{z}$ increases with D, making posterior inference much harder compared to the Bayesian logistic regression.

Prediction of a new input $\mathbf{x}_*$ can be obtained by integrating out the corresponding latent function z_* as shown in Eq. 1.25. The second term in the integral is the posterior distribution of z_*, derived by marginalizing out $\mathbf{z}$ from the joint distribution, as shown in Eq. 1.26 and 1.27.

$$p(y_* = 1|\mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}) = \int \sigma(z_*)p(z_*|\mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta})dz_* \quad (1.25)$$

$$p(z_*|\mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}) = \int p(z_*, \mathbf{z}|\mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta})d\mathbf{z} \quad (1.26)$$

$$= \int p(z_*|\mathbf{z}, \mathbf{x}_*, \mathbf{X}, \boldsymbol{\theta})p(\mathbf{z}|\mathbf{y}, \mathbf{X}, \boldsymbol{\theta})d\mathbf{z} \quad (1.27)$$

The first term is the conditional distribution of z_* given $\mathbf{z}$ and can be obtained using the fact that z_* and $\mathbf{z}$ are jointly Gaussian. This joint distribution is shown in Eq. 1.28 where Σ_{**} is the variance of z_* and $\boldsymbol{\Sigma}_*$ is the cross-covariance between z_* and $\mathbf{z}$ (note that $\boldsymbol{\Sigma}_*$ is a vector, but we denote it with a bold letter for simplicity). Finally, the conditional distribution is shown in Eq. 1.29.

$$p(z_*, \mathbf{z}|\mathbf{x}_*, \mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{z} \\ z_* \end{bmatrix} \middle| 0, \begin{bmatrix} \boldsymbol{\Sigma} & \boldsymbol{\Sigma}_* \\ \boldsymbol{\Sigma}_*^T & \Sigma_{**} \end{bmatrix}\right) \quad (1.28)$$

$$p(z_*|\mathbf{z}, \mathbf{x}_*, \mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}\left(z_*|\boldsymbol{\Sigma}_*^T\boldsymbol{\Sigma}^{-1}\mathbf{z}, \Sigma_{**} - \boldsymbol{\Sigma}_*^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_*\right) \quad (1.29)$$

The final task of interest is the computation of marginal likelihood,

$$p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = \int \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \prod_{d=1}^{D} e^{y_d z_d - \log(1+\exp(z_d))} d\mathbf{z} \tag{1.30}$$

Marginal likelihood is not only useful for model comparison, but also for optimizing the hyper-parameters $\boldsymbol{\theta}$. In GPC, classification accuracy is very sensitive to the setting of hyper-parameters since they encode the prior belief about correlation in the latent variables, making their estimation very important. We can jointly compute a distribution over $\mathbf{f}$ and $\boldsymbol{\theta}$ using MCMC, but a popular approach is to obtain a type II maximum likelihood estimate by maximizing the log of the marginal likelihood as shown below.

$$\hat{\boldsymbol{\theta}} = \arg\max_{\theta} \log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) \tag{1.31}$$

1.2.4 Probabilistic Principal Component Analysis

Principal component analysis (PCA) is a widely used technique for applications such as dimensionality reduction, visualization, and feature extraction. Given N data vector $\mathbf{y}_n$ with real-valued entries, PCA is a projection that minimizes the mean squared error between the data vectors and their projections [Pearson, 1901]. It can also be interpreted as the orthogonal projection of the data vectors onto a lower dimensional space such that the variance of the projection is maximized [Hotelling, 1933].

A probabilistic version of PCA was proposed by Tipping and Bishop [1999], where PCA is expressed as the maximum likelihood solution of a probabilistic latent variable model. There are several advantages of probabilistic PCA (PPCA), as discussed in Bishop [2006]. First of all, PPCA allows us to capture the correlation in the data with a fewer number of parameters, hence just like PCA it allows dimensionality reduction. An efficient EM algorithm can be obtained in cases where only few principal components are required [Ahn and Oh, 2003]. PPCA also allows for a principled approach to deal with missing values and to do model selection and comparison.

PPCA is a LGM. It assumes an isotropic, spherical Gaussian prior on the latent variables $\mathbf{z}_n$ and a factorial likelihood for $\mathbf{y}_n$, as shown in Eq. 1.32 and 1.33. Distribution for each dimension is a Gaussian distribution shown in Eq. 1.34. Here, σ^2 is the noise variance, $\mathbf{w}_d$ and w_{0d} are the factor

loading vector and offset respectively.

$$p(\mathbf{z}_n) = \mathcal{N}(\mathbf{z}_n|\mathbf{0}, \mathbf{I}) \tag{1.32}$$

$$p(\mathbf{y}_n|\mathbf{z}_n, \boldsymbol{\theta}) = \prod_{d=1}^{D} p(y_{dn}|\mathbf{z}_n, \boldsymbol{\theta}) \tag{1.33}$$

$$p(y_{dn}|\mathbf{z}_n, \boldsymbol{\theta}) = \mathcal{N}(y_{dn}|\mathbf{w}_d^T \mathbf{z}_n + w_{0d}, \sigma^2) \tag{1.34}$$

See Table 1.1 for equivalence between this model and LGM. The latent vector $\mathbf{z}_n$ contains the latent factors, and the observation y_{dn} can be modeled using the predictor $\mathbf{W}_d\mathbf{z} + \mathbf{w}_{0d}$. The table also gives the descriptions of various quantities such as N, D, L and K under this model.

There are many popular models similar to PPCA. Bayesian exponential family PCA (BxPCA) [Mohamed et al., 2008] is a generalization of PPCA to non-Gaussian data vectors and is also a special case of LGMs. BxPCA assumes a general exponential family likelihood instead of a Gaussian likelihood. For example, for binary data, we can use the Bernoulli logit likelihood, shown below.

$$p(y_{dn}|\mathbf{z}_n, \boldsymbol{\theta}) = \sigma(\mathbf{w}_d^T \mathbf{z}_n + w_{0d}) \tag{1.35}$$

A similar technique, called factor analysis (FA) [Bartholomew, 1980; Spearman, 1904], is also a special case of LGM. Here, the Gaussian likelihood in Eq. 1.33 is not spherical i.e. the noise covariance is a diagonal matrix with different variances along different dimensions.

A rather less popular model but yet important for this thesis is the latent Gaussian graphical model (LGGM). This model uses a Gaussian graphical model over latent variables to model the correlation in the data vectors. See Rue et al. [2009] and Yu et al. [2009] for few examples of this model. In its simplest form, we can assume that the number of latent variables is equal to the length of $\mathbf{y}_n$, i.e. $L = D$. The LGGM model then assumes a multivariate Gaussian prior over latent vectors as shown in Eq. 1.36. The likelihood of each dimension y_{dn} only depends on d'th entry of $\mathbf{z}_n$ as shown in Eq. 1.37. The parameters of this model are $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$.

$$p(\mathbf{z}_n|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{1.36}$$

$$p(\mathbf{y}_n|\mathbf{z}_n) = \prod_{d=1}^{D} p(y_{dn}|z_{dn}) \tag{1.37}$$

This model naturally arises in some applications, e.g. spatial statistics, where a sparse dependency structure on $\mathbf{z}_n$ is known [Rue et al., 2009].

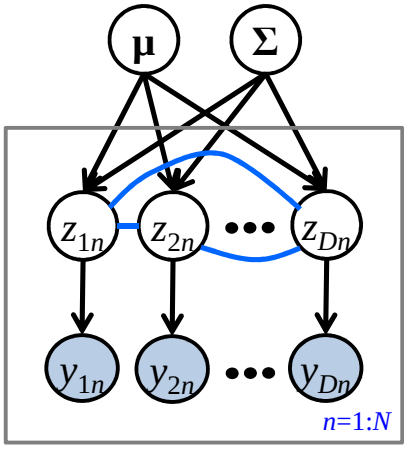


Figure 1.3: Latent Gaussian Graphical Model

However, it can also be used as a joint density model to explain the correlation in the observed data, e.g. see Yu et al. [2009] for application to collaborative filtering. The graphical model is shown in Fig. 1.3 which can be compared against the LGM graphical model in Fig. 1.1(b). See Table 1.1 for equivalence between this model and LGM. Note that, similar to GPC, there is one latent variable per observations, making the latent dimension equal to the data dimension.

We now discuss some tasks of interest for the PPCA model. We will discuss these in the context of binary data using likelihood shown in Eq. 1.35. First task of interest is the computation of posterior distribution of $\mathbf{z}_n$ given $\mathbf{y}_n$, shown below.

$$p(\mathbf{z}_n|\mathbf{y}_n, \boldsymbol{\theta}) \propto p(\mathbf{z}_n)p(\mathbf{y}_n|\mathbf{z}_n, \boldsymbol{\theta}) \tag{1.38}$$

$$= \mathcal{N}(\mathbf{z}_n|\mathbf{0}, \mathbf{I}) \prod_{d=1}^{D} e^{y_d(\mathbf{w}_d^T \mathbf{z}_n + w_{0d}) - \log(1+\exp(\mathbf{w}_d^T \mathbf{z}_n + w_{0d}))} \tag{1.39}$$

Again, the Gaussian distribution is not conjugate to the logistic function making the inference intractable.

Second task of interest is the computation of the marginal likelihood of

$\mathbf{y}_n$. This involves computation of an intractable integral as shown below.

$$p(\mathbf{y}_n|\boldsymbol{\theta}) = \int p(\mathbf{z}_n)p(\mathbf{y}_n|\mathbf{z}_n,\boldsymbol{\theta})d\mathbf{z}_n \tag{1.40}$$

$$= \int \mathcal{N}(\mathbf{z}_n|0,\mathbf{I}) \prod_{d=1}^{D} e^{y_d(\mathbf{w}_d^T\mathbf{z}_n+w_{0d})-\log(1+\exp(\mathbf{w}_d^T\mathbf{z}_n+w_{0d}))}d\mathbf{z}_n \tag{1.41}$$

The above marginal likelihood is also useful in the estimation of parameters $\boldsymbol{\theta}$, and fully Bayesian approaches based on MCMC can be applied. An easier approach, though, is to compute a point estimate of $\boldsymbol{\theta}$, for example, the type-II maximum likelihood estimate [Khan et al., 2010; Tipping, 1998; Tipping and Bishop, 1999].

Finally, we are interested in imputing missing values. Given a vector $\mathbf{y} = [\mathbf{y}^o \quad \mathbf{y}^m]$, where $\mathbf{y}^o$ is the observed part and $\mathbf{y}^m$ is the missing part, the probability that the i'th missing entry y_i^m takes a value 1 is given in Eq. 1.42. Here, we plug-in the estimate $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$, although a full distribution over $\boldsymbol{\theta}$ can also be used. As the equation suggests, we first compute the posterior distribution $p(\mathbf{z}|\mathbf{y}^o,\boldsymbol{\theta})$ and then approximate this integral with a Monte Carlo estimate.

$$p(y_i^m = 1|\mathbf{y}^o,\hat{\boldsymbol{\theta}}) = \int p(y_i^m = 1,\mathbf{z}|\mathbf{y}^o,\hat{\boldsymbol{\theta}})d\mathbf{z} \quad = \int p(y_i^m = 1|\mathbf{z},\hat{\boldsymbol{\theta}})p(\mathbf{z}|\mathbf{y}^o,\hat{\boldsymbol{\theta}})d\mathbf{z} \tag{1.42}$$

1.2.5 Correlated Topic Model

Topic models, such as latent Dirichlet allocation (LDA) [Blei et al., 2003], are useful in the analysis of documents. In LDA, for example, words in documents are represented as random mixtures over latent variables which can be interpreted as topics. The correlated topic model (CTM) [Blei and Lafferty, 2006] takes a step further and models the correlations among topics using a latent Gaussian vector. We now show that CTM is a special case of LGM.

We consider N documents with D words each with a vocabulary of size K. Let $\mathbf{z}_n$ be a length L real-valued vector for n'th document following a Gaussian distribution as shown in Eq. 1.43. Given $\mathbf{z}_n$, a topic t_{dn} is sampled for the d'th word in n'th document using a multinomial distribution shown in Eq. 1.44. Probability of that word y_{dn} is then given by Eq. 1.45. Here $\mathbf{B}$ is a $K \times L$ real-valued matrix with non-negative entries and rows that sum to 1, i.e. $\sum_{k=1}^{K} B_{kl} = 1$ for all l. The parameter set for this model is given

by $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{B}\}$.

$$p(\mathbf{z}_n|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{1.43}$$

$$p(t_{dn} = l|\mathbf{z}_n) = \frac{\exp(z_{ln})}{\sum_{j=1}^{L} \exp(z_{jn})} \tag{1.44}$$

$$p(y_{dn} = k|t_{dn}, \boldsymbol{\theta}) = B_{k,t_{dn}} \tag{1.45}$$

The graphical model is shown in Fig. 1.4(a).

We now show that this model is a LGM with a different likelihood function. Below, we derive this likelihood by marginalizing out the topic variables t_{dn}.

$$p(y_{dn} = k|\mathbf{z}_n, \boldsymbol{\theta}) = \sum_{l=1}^{L} p(y_{dn} = k|t_{dn} = l, \boldsymbol{\theta}) p(t_{dn} = l|\mathbf{z}_n) \tag{1.46}$$

$$= \sum_{l=1}^{L} B_{kl} \frac{\exp(z_{ln})}{\sum_{j=1}^{L} \exp(z_{jn})} \tag{1.47}$$

See Table 1.1 for equivalence between this model and LGM. The latent vector $\mathbf{z}_n$ is the topic vector using which the topic proportions are obtained. The words y_{dn} are then modeled using the topic proportions. The equivalence to LGM can be obtained by setting $\mathbf{W}_d$ and $\mathbf{w}_{0d}$ to the identity matrix and zero vector respectively, and using the likelihood above with an extra parameter $\mathbf{B}$. The complete parameter set is $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{B}\}$. The table also gives the descriptions of various quantities such as N, D, L and K under this model.

This model is closely related to the class of multinomial PCA (mPCA) model where the latent vector $\mathbf{z}$ is a probability vector [Buntine, 2002]. For example, LDA, a member of mPCA class, assumes a Dirichlet prior over $\mathbf{z}$. As a result, $\mathbf{z}$ lies in the probability simplex with non-negative elements which sum to 1. Since $\mathbf{z}$ is probability vector, the topic variable t_{dn} can directly be modeled with a multinomial distribution. In CTM, this required the use of the multinomial logit likelihood which first transforms the real-valued vector $\mathbf{z}$ to a probability vector and then use multinomial distribution to model the topics. We can see in Fig. 1.4(b) that the graphical model of LDA is very similar to that of CTM. The advantage of CTM, however, is that it can model correlations between topics using the covariance matrix $\boldsymbol{\Sigma}$ [Blei and Lafferty, 2006]. The topic models can also be combined with the PCA type models to model the correlation between documents and other types of variables, e.g. see the ideal point model of Gerrish and Blei [2011].

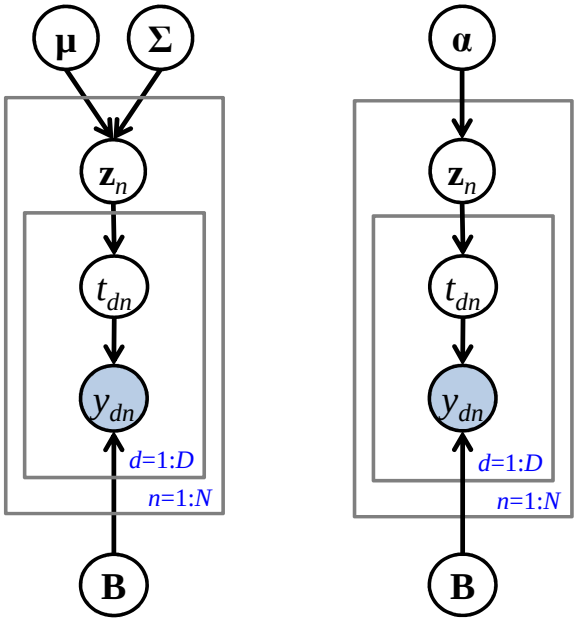


Figure 1.4: (a) Correlated topic model vs (b) latent Dirichlet allocation

The tasks of interest in CTM are very similar to what we discussed before for the PPCA and discrete choice models. The difficulty in these tasks arises from the non-conjugacy of the likelihood shown in Eq. 1.47 to the Gaussian prior.

1.3 Distributions for Discrete Observations

We now discuss choices of the distribution $p(y_{dn}|\boldsymbol{\eta}_{dn})$ for various types of discrete measurements. Specifically, we consider the following four types: binary, count, ordinal, and categorical. For notational simplicity, we will drop the subscript d and n, and specify the distribution $p(y|\boldsymbol{\eta})$ for a scalar observation y with predictor $\boldsymbol{\eta}$.

There are two popular approaches to derive distributions for discrete observations: the random utility model (RUM) [Marschak, 1960] and the generalized linear model (GLM) [McCullagh and Nelder, 1989]. The former is more popular in statistics and bio-statistics, while the later is popular in econometrics and psychometrics. Although the two approaches are fundamentally different, they can give rise to equivalent models [Skrondal and

Link Function	$g(\mu)$
Logit	$\log(\mu/(1-\mu))$
Probit	$\Phi^{-1}(\mu)$
Logarithm	$\log(\mu)$
Complementary log-log	$\log(-\log(1-\mu))$

Table 1.2: Link functions for the generalized linear model. Here, $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution.

Rabe-Hesketh, 2004].

In the RUM approach or otherwise known as the latent response model (LRM), we consider a latent utility $u = \eta + \epsilon$, where ϵ is an error term. An observation is then generated from the utility using different kinds of threshold functions. A GLM is defined by two components. First, a link function $g(\cdot)$ which links the expectation of the observation (denoted by μ) to the linear predictor η as $\eta = g(\mu)$. Examples of g are shown in Table 1.2. Second, a conditional probability distribution of the observation is chosen from the exponential family distribution. For a scalar observation y with a scalar predictor η, the exponential family distribution is defined as follows,

$$p(y|\eta) = h(y) \exp\left(\eta T(y) - A(\eta)\right) \tag{1.48}$$

Here, $T(y), h(y)$, and $A(\eta)$ are known functions. We now describe distributions for different types of observations derived using the two models.

1.3.1 Binary Observations

We model a binary (or dichotomous) observation $y \in \{0, 1\}$ with the Bernoulli distribution $p(y = 1|\mu) = \mu$, where μ is the expectation of the binary variable which can be modeled using logit and probit links as shown in Eq. 1.49 and 1.50.

$$\mu = \frac{\exp(\eta)}{1 + \exp(\eta)} = \sigma(\eta) \tag{1.49}$$

$$\mu = \Phi(\eta) \tag{1.50}$$

These distributions can also be derived using the RUM approach. Assuming a normal distribution for the error term gives rise to the probit model, while assuming a logistic cumulative distribution function for the error term gives rise to the logit model (we will discuss this in detail in Section 1.3.3).

1.3.2 Count Observations

Count observations take non-negative integer values $\{0, 1, 2, 3, \dots, \}$. Such an observation can be modeled using the Poisson distribution with expectation μ, given in Eq. 1.51. We link the expectation μ to the predictor η using a log link where $\mu = e^{\eta}$.

$$p(y = k|\eta) = \frac{e^{-\mu}\mu^k}{k!} \tag{1.51}$$

1.3.3 Categorical Observations

A categorical observation takes value in a discrete set $\{C_0, C_1, \dots, C_K\}$, where each C_k is a category. This type of observation is also referred to as multinomial, nominal, polychotomous, quantal, or discrete choices. To model categorical observations, we need to define a separate linear predictor η_k for $k = 0, 1, \dots, K$. We denote the vector of η_k's by $\boldsymbol{\eta}$.

We derive the distribution $p(y = C_k|\boldsymbol{\eta})$ using the random utility model (RUM) framework, defining utilities as $u_k = \eta_k + \epsilon_k, \forall k$ where ϵ_k is the error term [Marschak, 1960]. In this framework, we choose $y = C_k$ if k'th category has the highest utility, i.e. when $u_k > u_j, \forall k \neq j$. The probability of this choice depends on the probability distribution of the random errors. Specifically, the probability of $y = C_k$ is equal to the probability of the region where $u_k > u_j, \forall j \neq k$ as shown in Eq. 1.52. This can be expressed in terms of the probabilities of η_k's as shown in Eq. 1.53 and 1.54.

$$\begin{aligned} p(y = C_k|\eta) &= \mathbb{P}\left(u_k > u_j, \forall j \neq k\right) && (1.52)\\ &= \mathbb{P}\left(\eta_k + \epsilon_k > \eta_j + \epsilon_j, \forall j \neq k\right) && (1.53)\\ &= \mathbb{P}\left(\epsilon_j < \epsilon_k + \eta_k - \eta_j, \forall j \neq k\right) && (1.54) \end{aligned}$$

This is a K dimensional integral but in some cases a closed form expression can be obtained. The only thing left to do is to define the distribution of the random errors.

If we assume ϵ_k to be independently and identically distributed according to the Gumbel distribution, we obtain the multinomial logit distribution as shown below.

$$p(y = C_k|\boldsymbol{\eta}) = \frac{e^{\eta_k}}{\sum_{j=0}^{K} e^{\eta_j}} \tag{1.55}$$

This expression can be derived using the fact that the cumulative distribution function of the Gumbel distribution is $F(\epsilon_k) = \exp(-\exp(-\epsilon_k))$ (see

Train [2003] for details). There are many variants of the logit distribution depending on the construction of the predictor, for example, in the conditional logit model [McFadden, 1973], regression weights do not depend on data examples [Hoffman and Duncan, 1988]. In this thesis, we do not consider these special cases but our results can be trivially extended to them.

If errors follow a multivariate normal distribution $p(\boldsymbol{\epsilon}) = \mathcal{N}(0, \boldsymbol{\Omega})$, we get the multinomial probit distribution. There is no closed form expression for this distribution in general. However, if we assume that the ϵ_k's are iid standard normal, we get a simplification of the K dimensional integral to a one-dimensional integral shown below. This is a very common assumption made in machine learning community, for example, see Girolami and Rogers [2006]; Seeger et al. [2006].

$$p(y = C_k|\boldsymbol{\eta}) = \int \prod_{j \neq k} \Phi(u + \eta_k - \eta_j) du \tag{1.56}$$

Both the logit and probit distributions have been extensively used in the literature, see for example, Albert and Chib [1993]; Chib and Greenberg [1998]; Frühwirth-Schnatter and Frühwirth [2010]; Frühwirth-Schnatter and Wagner [2008]; Girolami and Rogers [2006]; Holmes and Held [2006]; Scott [2011]. Although both distributions tend to give similar goodness of fit and qualitative conclusions [Holmes and Held, 2006; Train, 2003], they both have their own advantages and disadvantages. One of the advantages of logit over probit is that its parameters are interpretable: for two categories k and l, the log of ratio of probabilities is equal to the difference between their predictors, as shown below.

$$\log \frac{p(y = C_k|\eta)}{p(y = C_l|\eta)} = \log \frac{e^{\eta_k} / \sum_j e^{\eta_j}}{e^{\eta_l} / \sum_j e^{\eta_j}} = \log \frac{e^{\eta_k}}{e^{\eta_l}} = \eta_k - \eta_l \tag{1.57}$$

Another advantage of the logit distribution is that it has slightly fatter tails than the probit distribution, which means that the logit distribution allows for slightly more aberrant behavior than the probit distribution. Fatter tails are due to the use of extreme value distribution in the logit distribution, instead of the normal distribution used in the probit distribution.

A disadvantage of logit over probit is that it makes an assumption of "independence of irrelevant alternatives (IIA)". Note from Eq. 1.57 that the ratio of probabilities of two categories is independent of any other category, or in other words any "irrelevant" category. The IIA property arises from the assumption that the error terms are independent, an assumption which may not always hold. For example, a person who dislikes traveling by bus because

of the presence of other travelers might have a similar reaction to train travel. In this situation, the unobserved factors, related to the "presence of other travelers", are correlated for the bus and train categories rather than being independent. For a detailed discussion of the IIA property and other examples, see Chapter 3 of Train [2003]. The IIA property, however, can be useful when it holds in reality. For example, in some situations, we can reduce the computation by selecting subsets of categories, and since their probabilities are independent of other categories, their exclusion does not affect the analysis. The probit distribution does not have the IIA property due to the presence of a correlated noise, but the independent noise model (shown in Eq. 1.56) indeed exhibit the IIA property and has similar issues as the logit distribution.

Both the logit and probit model have identifiability issues, the probit model having more serious problems. A parameter of the model is *identified* if it can be estimated and is *unidentified* otherwise. To understand the problem, note that in a random utility model, the level and scale of the utilities are irrelevant. The level of utilities does not matter because a constant can be added to the utility of all the categories without changing the category with maximum utility. Similarly, multiplying utilities with a positive constant does not change the category with maximum utility. It is possible that some parameters of the distribution might relate to the scale and level of utility and do not affect the behavior of the distribution. For example, a parameter b in utilities $u_k = \eta_k + b + \epsilon_k$ is unidentified since the probabilities do not depend on it. The difficulty arises because it is not always obvious which parameters relate to scale and level. The logit model has an advantage since the unidentifiability can be dealt with by simply fixing one of the predictor to zero. The solution is simple since the independence and constant variance assumption on the error terms already removes all the unidentifiability problem associated with the scale and level. For the probit model, however, the solution is not that simple since the errors are correlated and might affect scale and level in a complicated way. Bunch [1991] discusses several published articles that considered probit models with unidentifiability problem, perhaps unknowingly. Such is the complication associated with this issue. Train [2003] suggests a normalization procedure that involves reparameterizing $\mathbf{\Omega}$ and setting the first element of the matrix to 1. Several other normalization procedures are discussed in Bunch [1991]. It is worth noting that the probit model of Eq. 1.56 restricts the error terms to be independent and the identifiability issue can be dealt in a similar way as the logit distribution.

There exist other distributions such as generalized extreme value distri-

butions and mixed logit which contain many more flavors of distributions eliminating the disadvantages of logit and probit distributions; see Train [2003] for details. For the purpose of this thesis, the multinomial logit and probit distributions already pose many challenging estimation problem. Therefore, we focus only on these distributions.

1.3.4 Ordinal Observations

An ordinal observation takes value in an ordered discrete set $\{0, 1, \ldots, K\}$. We can define distributions for an ordinal observation y by linking the cumulative probability $\mathbb{P}(y \leq k|\eta)$ to the predictor using a GLM, as shown in Eq. 1.58. Here, ϕ_k's are real threshold parameters such that $-\infty = \phi_0 < \phi_1 < \ldots < \phi_K = \infty$. We can use logit or probit links as shown in Eq. 1.59.

$$\mathbb{P}(y \leq k|\eta) = g^{-1}(\phi_k - \eta), k = 0, 1, 2, \ldots, K \tag{1.58}$$

$$= \begin{cases} \Phi(\phi_k - \eta), & \text{for probit link} \\ \sigma(\phi_k - \eta), & \text{for logit link} \end{cases} \tag{1.59}$$

This family of models is referred to as the cumulative model since we use the cumulative probabilities to derive the model. These models are also known as graded response models [Samejima, 1997]. The model with the probit link is called the ordered probit model, while the logit link model is known as the cumulative logit model. These models are also called the proportional odds model since the odds of two ordinal observations is same for all categories [McCullagh, 1980]. To show this we first note that the ratio of the probability of $y \leq k$ to the probability that $y > k$ is proportional to η.

$$\log\left[\frac{\mathbb{P}(y \leq k|\eta)}{\mathbb{P}(y > k|\eta)}\right] = \log\frac{\sigma(\phi_k - \eta)}{1 - \sigma(\phi_k - \eta)} = \log\frac{e^{\phi_k - \eta}/(1 + e^{\phi_k - \eta})}{1/(1 + e^{\phi_k - \eta})} = \phi_k - \eta \tag{1.60}$$

Given two ordinal observations y_i and y_j with predictors η_i and η_j, the odds of this ratio will be proportional to $\eta_j - \eta_i$, which is same for all the categories, hence the name proportional odds model. This proportionality property may not always hold (see Ananth and Kleinbaum [1997] for an example). However, an important feature of these models is that they are invariant under the collapsability of the categories, i.e. if two adjacent categories are collapsed, predictor values do not change (although thresholds are affected). Also, if the categories are reversed, the model remains unaffected.

An alternative model, called the continuation ratio model, uses the logit link and assumes the following,

$$\log\left[\frac{\mathbb{P}(y=k|\eta)}{\mathbb{P}(y>k|\eta)}\right] = \phi_k - \eta \tag{1.61}$$

When the logit link is replaced by the complimentary log-log link, the resulting model is the Cox proportional-hazard model for survival data in discrete time. See McCullagh and Nelder [1989] for a detailed discussion. The continuation ratio model is suited for the cases where the categories are not merely an arbitrary grouping of an underlying continuous variable. However, unlike the proportional hazard model, this model is not invariant under the collapsability and reversal of the categories.

The cumulative model assumes the existence of a one-dimensional predictor function which is thresholded to get ordinal observations. In many datasets, this may not hold. Anderson [1984] gives two specific scenarios where this may not be the case. First concept is related to the existence of a multi-dimensional predictor function rather than a one-dimensional one. For example, to assess the severity of a disease, a physician might use different kinds of tests depending on the severity, making the use of a one-dimensional predictor inappropriate. Another related concept is of "indistinguishability". It might happen that a predictor can be used to distinguish between two categories, but may not be predictive of others. In this case, again, use of a one-dimensional predictor function is not valid.

Anderson [1984] proposed the stereotype regression model which is a restriction of the multinomial logit model of Section 1.3.3. The multinomial logit model has $K+1$ predictors, one per category. Stereotype model restricts the form of these predictors to a linear function of a single predictor η by defining them as $\eta_k = \phi_k - \alpha_k\eta$, resulting in the model shown below.

$$p(y=k|\eta) = \frac{\exp(\phi_k - \alpha_k\eta)}{\sum_{j=0}^{K}\exp(\phi_j - \alpha_j\eta)} \tag{1.62}$$

We assume $1 = \alpha_0 \geq \alpha_1 \geq \ldots \geq \alpha_K = 0$ to make use of the ordering constraints and to make the model identifiable. A simplification of the stereotype model is obtained by assuming that $\alpha_j = j$, giving rise to the adjacent category model [Agresti, 2010].

The parameters of the stereotype model have a simple intuitive interpretation. The log of ratio of the probabilities of two categories can be expressed as shown in Eq. 1.63, which shows that the odds of having response k instead of j is influenced heavily by predictors, besides the difference $\phi_k - \phi_j$,

only if the difference $\alpha_j - \alpha_k$ is large. The effect of predictors grows stronger with the separation between the categories, which is a desirable property to have.

$$\log \frac{p(y=k|\eta)}{p(y=j|\eta)} = \phi_k - \phi_j + (\alpha_j - \alpha_k)\eta \tag{1.63}$$

The stereotype model solves the problem of multi-dimensional predictor and indistinguishability of the categories. The model can handle the multi-dimensional predictor function. Anderson gives an example of this in two dimensions where the predictor for k'th category can take a form $\alpha_{k1}\eta_1 - \alpha_{k2}\eta_2$ where η_1 and η_2 are two different predictors. This can be generalized to higher dimensions. We can also easily check the assumption of indistinguishability, for example, hypothesis of the form $\alpha_k = \alpha_j$ can be used to check if two categories are indistinguishable or not.

1.4 Learning Objectives

In Section 1.2, we discussed many popular models and the learning objectives associated with them. In this section, we summarize those problems in the context of the generic LGM. There are four main tasks that we are interested in this thesis, listed below. Our goal is to perform these task accurately and efficiently.

Posterior inference The first task of interest is the inference of posterior distribution over $\mathbf{z}_n$ given $\mathbf{y}_n$ and $\boldsymbol{\theta}$, shown below.

$$p(\mathbf{z}_n|\mathbf{y}_n, \boldsymbol{\theta}) \propto \mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \prod_{d=1}^{D} p(y_{dn}|\mathbf{z}_n, \boldsymbol{\theta}) \tag{1.64}$$

This task arises in almost all LGMs discussed in Section 1.2. The posterior distribution does not have a closed form expression, since $p(y_{dn}|\mathbf{z}_n, \boldsymbol{\theta})$ are not conjugate to the Gaussian prior over $\mathbf{z}_n$.

Marginal likelihood The second task of interest is the estimation of the log-marginal likelihood of $\mathbf{y}_n$ given $\boldsymbol{\theta}$.

$$\mathcal{L}_n(\boldsymbol{\theta}) := \log p(\mathbf{y}_n|\boldsymbol{\theta}) = \int \mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \prod_{d=1}^{D} p(y_{dn}|\mathbf{z}_n, \boldsymbol{\theta}) d\mathbf{z}_n \tag{1.65}$$

This is useful for model selection. It is also useful to compute an estimate of parameters as discussed in our next objective. This integral is intractable because of non-conjugacy and our goal is to compute an approximation to the log-marginal likelihood.

Parameter estimation The third task of interest is the estimation of $\boldsymbol{\theta}$ given data vectors $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$. In this thesis, we focus on finding a maximum likelihood estimate, defined below.

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) := \sum_{n=1}^{N} \mathcal{L}_n(\boldsymbol{\theta}) \tag{1.66}$$

This task is useful in Gaussian process classification and almost all latent factor models discussed in Section 1.2. Instead of a point estimate, we might want to infer the posterior distribution of $\boldsymbol{\theta}$ which is a much more difficult task. We focus on a point estimate since this problem is challenging enough in itself.

Prediction The final task of interest is the prediction of unobserved (or missing) part of a data vector given $\boldsymbol{\theta}$. For a vector $\mathbf{y} = [\mathbf{y}^o\ \mathbf{y}^m]$, where $\mathbf{y}^o$ is observed, we compute a prediction of the i'th missing entry y_i^m as shown below,

$$p(y_i^m|\mathbf{y}^o, \boldsymbol{\theta}) = \int_z p(y_i^m, \mathbf{z}|\mathbf{y}^o, \boldsymbol{\theta}) d\mathbf{z} = \int_z p(y_i^m|\boldsymbol{\eta}) p(\mathbf{z}|\mathbf{y}^o, \boldsymbol{\theta}) d\mathbf{z} \tag{1.67}$$

This task is useful for filling in the missing values in the factor models such as PCA. The task of prediction in the regression models can also be derived as a special case of this task.

1.5 Summary of Contributions

Below, we briefly summarize the contributions made in each chapter.

In **Chapter 2**, we review existing methods for learning in discrete-data LGMs. We show that the source of difficulty in learning is the non-conjugacy of discrete data likelihoods to the Gaussian prior. We review solutions which can be categorized in three major categories: non-Bayesian methods, sampling methods, and deterministic methods. Every solution comes with their own advantages and disadvantages, which we discuss thoroughly, showing

inadequacy of these methods.

In **Chapter 3**, we discuss the main approach used in this thesis: variational learning using the *evidence lower bound optimization*. We show that the lower bound is not tractable for discrete-data likelihoods and illustrate the use of local variational bounds (LVBs) to achieve tractability. We devote the rest of the chapter to improve computational aspects of the lower bound. We make three contributions in this regard. Firstly, we establish the concavity of the variational lower bound. Secondly, we derive the generalized gradient expressions for its optimization. Finally, we propose a fast convergent variational algorithm for special LGMs, such as Gaussian processes. Last contribution is based on Khan et al. [2012b].

In **Chapter 4**, we focus on tractable variational learning for binary data. We consider the Bernoulli logit LGM for which learning is intractable. We show that the existing solution, the bound proposed by Jaakkola and Jordan [1996], can lead to slow and inaccurate learning. We make two contributions. First, we propose use of the Bohning bound which leads to faster, but less accurate, learning algorithm than the Jaakkola bound. Second, we derive new *fixed, piecewise linear and quadratic bounds*. These bounds have bounded maximum error which can be driven to zero by increasing the number of pieces, giving rise to variational algorithms with a wide-range of speed accuracy trade-offs. These bounds are more accurate than the Jaakkola bound, but have the same computational cost. This chapter is based on Marlin et al. [2011].

In **Chapter 5**, we focus on tractable variational learning for categorical data. We first consider the multinomial logit LGM and review existing LVBs for it. In our first contribution, we extend the use of the Bohning bound and show that it leads to a fast variational algorithm. Unfortunately, our error analysis reveals that all the LVBs for multinomial logit LGM, including the Bohning bound, can be inaccurate at times, and unlike the binary case, designing bounds with error guarantees is difficult. We take another approach to solve the problem. In our second contribution, we propose a new likelihood called the *stick breaking* likelihood. The advantage of the new likelihood is the availability of accurate LVBs. These contributions are based on Khan et al. [2010] and Khan et al. [2012a].

In **Chapter 6**, we present some extensions and discuss future work. We extend our variational approach to ordinal and mixed-data LGMs. Mixed-

data results are based on Khan et al. [2010]. We discuss future work on improving the computational efficiency of our approach, as well as extending our approach to other likelihoods than discrete-data likelihoods. In **Chapter 7**, we summarize our conclusions.

Chapter 2

Learning Discrete-Data LGMs

The difficulty in learning discrete-data LGMs lies in the non-conjugacy of discrete-data likelihoods to the Gaussian prior. In this chapter, we review existing methods to solve the problem. These methods can be categorized in three major categories: non-Bayesian methods, sampling methods, and deterministic methods. For each category, we briefly review the basic methodology and discuss its advantages and disadvantages, demonstrating the method's insufficiency to provide satisfactory solutions. To summarize, non-Bayesian approaches are fast but can be inaccurate, while sampling methods are slower but more accurate. Deterministic methods provide a good alternative to these methods, providing better accuracy than non-Bayesian methods in less time than sampling methods, but are less general.

2.1 Non-Bayesian Approaches

Instead of computing a full posterior distribution over latent variables, non-Bayesian approaches compute a point estimate. For example, a popular choice is the maximum-a-posteriori (MAP) estimate, which can be obtained by maximizing the posterior distribution, as shown below.

$$\hat{\mathbf{z}}_n = \arg\max_{\mathbf{z}_n} \log p(\mathbf{z}_n|\mathbf{y}_n, \boldsymbol{\theta}) = \arg\max_{\mathbf{z}_n} \log p(\mathbf{y}_n|\mathbf{z}_n, \boldsymbol{\theta}) p(\mathbf{z}_n|\boldsymbol{\theta}) \tag{2.1}$$

The advantage of this approach is that the maximization can be performed efficiently, since the objective function is concave for most of the likelihoods.

Non-Bayesian approaches are computationally efficient, not only in computing the point estimate but also in all other learning tasks. For example, the marginal likelihood can be estimated simply by plugging in the MAP estimate in the joint likelihood. Furthermore, given the MAP estimate, parameters can be obtained by maximizing the marginal likelihood approx-

imation, as shown below.

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \sum_{n=1}^{N} \log p(\mathbf{y}_n|\mathbf{z}_n^*, \boldsymbol{\theta}) p(\mathbf{z}_n^*|\boldsymbol{\theta}) \tag{2.2}$$

For most of the likelihoods, the approximation is concave with respect to $\boldsymbol{\theta}$, making the above optimization efficient. This leads to a simple coordinate ascent approach for parameter estimation, which involves iterating between solving Eq. 2.1 and 2.2 respectively. An example of this approach is the exponential family PCA (EPCA) [Collins et al., 2002; Kabán and Girolami, 2001; Orlitsky, 2004; Salakhutdinov and Mnih, 2008b], where sometimes it is even possible to achieve the global maximum [Guo and Schuurmans, 2008].

Non-Bayesian approaches are simple, efficient, and widely applicable, but these benefits come at the expense of ignoring the posterior variance of $\mathbf{z}$. Welling et al. [2008] argue that, for models with continuous latent variable, the non-Bayesian approaches "optimize a questionable objective". They show that the marginal likelihood is lower bounded by the cost function of Eq. 2.2 plus the entropy of a Dirac-delta distribution which is 1 at the MAP estimate and zero otherwise. The entropy term is $-\infty$ for continuous latent variables and ignoring it can potentially make the resulting objective function numerically unrelated to the marginal likelihood. Although this may not always lead to bad parameter estimates in practice, Welling et al. [2008] suggest to be careful about the performance evaluation of non-Bayesian methods.

A more practical problem with non-Bayesian methods is related to overfitting and their sensitivity to the regularization parameter [Khan et al., 2010; Mohamed et al., 2008; Salakhutdinov and Mnih, 2008a,b]. We illustrate this effect using a simulation experiment. We generate a continuous dataset using $D = 10$, $L = 5$, and $N = 200$ data cases by sampling from the factor analysis (FA) model described in Section 1.2.4. We fix $\sigma^2 = 0.1$ and $w_{0d} = 0$, while estimating $\mathbf{w}_d$. We consider the case of 10% and 50% missing data. We compare the non-Bayesian approach with two Bayesian approaches. The first approach is the fully Bayesian approach, based on the hybrid Monte Carlo (HMC) algorithm [Mohamed et al., 2008]. This method samples both $\boldsymbol{\theta}$ and $\mathbf{z}$, hence we call it the sample-sample (SS) approach. The second approach is the variational approach, based on the expectation maximization (EM) algorithm [Tipping and Bishop, 1999]. This approach obtains a full posterior distribution over $\mathbf{z}$ using the variational lower bound, but only computes a point estimate for $\boldsymbol{\theta}$. Hence, we refer to this approach as the variational-maximize (VM) approach. Finally, we refer to the non-

Bayesian approach as the maximize-maximize or MM approach.

Our goal is to illustrate the sensitivity of the non-Bayesian approach to regularization parameter. For this purpose, we use a l_2 regularizer for $\mathbf{W}$ and denote its coefficient by λ_w. We evaluate the sensitivity of the methods to λ_w by varying it over the range 10^{-2} to 10^2. We first randomly split the data in 50/50 train/test split, and train the methods on the observed entries in the training set. Then, we compute mean-square-error (MSE) on the missing entries in the training and test sets. Finally, we average the results over 20 different data splits and plot the average against λ_w.

Top row in Fig. 2.1 shows that the test MSE of the non-Bayesian method is extremely sensitive to the prior precision λ_w. We can see that this sensitivity increases with an increase in the missing data rate. We hypothesize that this is a result of the non-Bayesian method ignoring the posterior uncertainty in $\mathbf{z}$. This is supported by looking at the MSE on the training set in the bottom right plot. We see that the non-Bayesian method overfits when λ_w is small. Consequently, it requires a careful discrete search over the values of λ_w, which is slow, since the quality of each such value must be estimated by cross-validation. By contrast, both the Bayesian methods take the posterior uncertainty in $\mathbf{z}$ into account, resulting in almost no sensitivity to λ_w over this range.

Another advantage of Bayesian approaches over non-Bayesian methods is that they easily handle the hierarchical models, such as choice models and correlated topic models.

2.2 Sampling Methods

An alternative approach is to obtain samples using methods based on Markov Chain Monte Carlo (MCMC) sampling. MCMC offers a variety of algorithms which are general, flexible, widely applicable, and many times easy to implement and parallelize. For this reason, MCMC methods have been applied quite extensively to latent Gaussian models and are shown to perform well. In fact, MCMC can be considered a gold standard for problems of moderate dimensionality.

The main problem with MCMC is that they usually exhibit slow *mixing* leading to a slow exploration of the parameter space, especially for high dimensional data. This becomes even a bigger problem in practice since it is difficult not only to predict the *convergence* of MCMC algorithms but also to diagnose or detect it. For this matter, most of the theoretical convergence results for MCMC are of little practical use. Presence of algorithmic

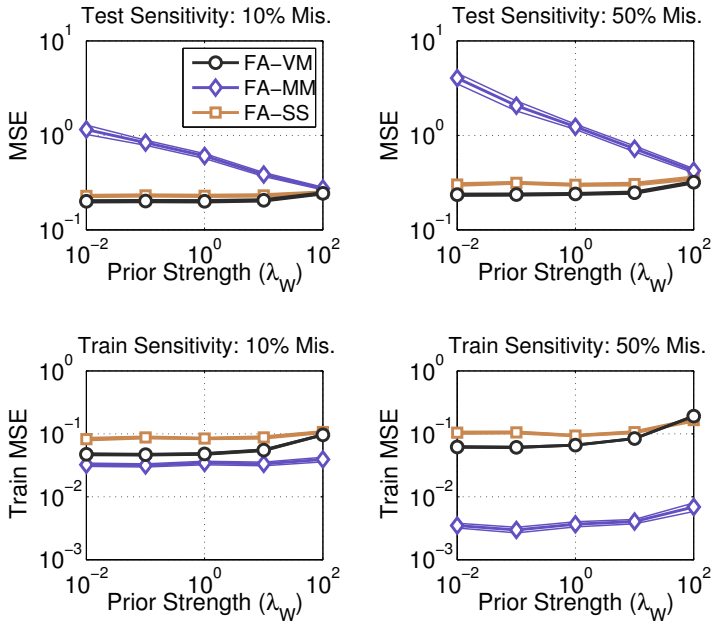


Figure 2.1: MSE vs λ_w for MM, VM, and SS approaches for the FA model. We show results on the test and training sets with 10% and 50% missing data. Top row shows that the test MSE of the non-Bayesian method is extremely sensitive to the prior precision λ_w, while the bottom right plot shows its overfitting.

parameters makes their use cumbersome, and sometimes lead to sub-optimal performances. These problems are even more severe in high dimensions, prohibiting use of MCMC to large-scale problems. In addition, MCMC methods have special difficulty in estimation of marginal likelihood which remains an open research problem. In summary, MCMC methods do offer a solution to all of our objectives but they come with their own problems and issues. In what follows, we discuss these issues in detail for each task of interest.

2.2.1 Posterior Inference

It is difficult to sample directly from $p(\mathbf{z}|\mathbf{y}, \boldsymbol{\theta})$ for at least three reasons. Firstly, we do not know the normalizing constant of this distribution. Secondly, the distribution does not take a convenient parametric form that is easy to sample from. Finally, the latent vector $\mathbf{z}$ might be high-dimensional, making sampling difficult. The good news is that it is easy to evaluate the

unnormalized distribution $\tilde{p}(\mathbf{z}) = p(\mathbf{y}|\mathbf{z}, \boldsymbol{\theta})p(\mathbf{z}|\boldsymbol{\theta})$ which motivates the use of MCMC methods.

MCMC methods generate samples from a "hard-to-sample" distribution by constructing a Markov process whose *invariant* distribution is the desired distribution. The most popular algorithm is the well-known *Metropolis-Hastings* (MH) algorithm, where we first propose a new sample $\mathbf{z}^*$ given an old sample $\mathbf{z}$ using a *proposal* distribution $q(\mathbf{z}^*|\mathbf{z})$, and then accept the new sample with the following probability.

$$p(\mathbf{z}, \mathbf{z}^*) = \min\left[1, \frac{\tilde{p}(\mathbf{z}^*)q(\mathbf{z}|\mathbf{z}^*)}{\tilde{p}(\mathbf{z})q(\mathbf{z}^*|\mathbf{z})}\right] \tag{2.3}$$

The Markov chain formed from this procedure has the desired posterior distribution as its equilibrium, and eventually the samples are drawn from the posterior and the sampler is *converged*. Another popular MCMC algorithm is Gibbs sampling which is a special case of MH algorithm with proposal distribution defined in terms of the conditionals of $p(\mathbf{z}|\mathbf{y}, \boldsymbol{\theta})$. This is useful when it is easier to sample from the conditionals instead of the joint.

Standard MCMC algorithms have been extensively applied for inference in discrete-data LGMs, for example, Gibbs sampling [Zeger and Karim, 1991], single move adaptive rejection Gibbs sampling [Dellaportas and Smith, 1993], Metropolis Hasting algorithm [Gamerman, 1997; Lenk and DeSarbo, 2000; Rossi and Allenby, 2003]. These algorithms usually suffer from slow mixing and lead to very lengthy simulations. For example, if the variables are highly correlated, Gibbs sampling mixes slowly since samples from conditional are correlated as well. Similarly, MH algorithms that are based on random walk proposal distributions need to make smaller steps to achieve higher acceptance rates since larger steps will be rejected more often. This leads to poor exploration of the state-space and slow mixing. See MacKay [2003] for details on problems with MCMC.

There are several approaches to improve the mixing of MCMC algorithms of which the following two classes of algorithms are popular for discrete-data LGM: the Hamiltonian Monte Carlo (HMC) algorithm [Duane et al., 1987] and the data-augmentation approach [Albert and Chib, 1993]. We briefly discuss each of these approaches below.

Hamiltonian Monte Carlo algorithm

Since its first introduction in the machine learning community by Neal [1992], HMC has been a popular choice, e.g. for Gaussian process classification [Kuss and Rasmussen, 2005; Nickisch and Rasmussen, 2008] and for

Bayesian exponential family PCA [Mohamed et al., 2008]. HMC defines a Hamiltonian function in terms of the posterior distribution by introducing auxiliary variables called the momentum variables which typically have Gaussian distributions. To generate a new sample from the posterior distribution, we proceed as follows: first, sample momentum variables, then propose a new state by computing a trajectory according to the Hamiltonian dynamics, and finally use a MH step to either accept or reject this new sample. First and third steps are easy since momentum variables follow a Gaussian distribution and the posterior distribution is easy to evaluate. Second step is the difficult one since it requires a discretization scheme for implementation of the Hamiltonian dynamics. Usually the *leapfrog* scheme is employed which has two important parameters that must be set by hand, namely the step size and the number of leapfrog steps [Neal, 1992].

Mixing of HMC sampler is very sensitive to the setting of these parameters making implementation of HMC difficult in practice; see Neal [1992] for a detailed discussion. For example, too large a step size results in a very low acceptance rate due to the error introduced in Hamiltonian dynamics computation, while too small a step-size will waste computation or lead to a slow exploration of the state-space. In a recent work, Girolami and Calderhead [2011] suggest an extension of HMC that reduces the sensitivity to the parameters. They use a 'position-specific' covariance parameter $G(\mathbf{z})$ for the momentum prior based on the Riemann manifold, naming their method Riemann manifold HMC. Although, this does indeed improve the mixing of HMC, it also increases the computation cost heavily since the Hamiltonian computation now involves inversion of $G(\mathbf{z})$ for every new $\mathbf{z}$ inside the leapfrog scheme. A recent modification, called the No-U-Turn Sampler (NUTS), eliminates the need to set the number of leapfrog steps [Hoffman and Gelman, 2011]. This modification seems promising and can be improved further; see Radford Neal's comments at his blog[2].

The data augmentation approach

The second class of algorithms are based on data augmentation, also known as the auxiliary variable approach. Examples of these are the data augmented Gibbs sampling [Albert and Chib, 1993; Frühwirth-Schnatter and Frühwirth, 2010; Holmes and Held, 2006] and data augmented Metropolis-Hastings sampler [Scott, 2011]. These methods make use of the random utility model (RUM) of the probit and logit links to introduce auxiliary

[2] `http://radfordneal.wordpress.com/2012/01/21/`

variables (see Section 1.3 for details on RUM). Interested readers should see Van Dyk and Meng [2001] for examples where data augmentation shows faster mixing than the standard MCMC algorithm. Although these methods do mix faster than the standard MCMC algorithms, they still suffer from convergence diagnostics issues. Most of the theoretical convergence results for MCMC are of little practical use, and diagnosing the convergence of MCMC algorithms takes expert knowledge [MacKay, 2003].

2.2.2 Marginal Likelihood Estimation

In this section, we discuss estimation of marginal likelihood using MCMC.

The harmonic mean estimator

The harmonic mean estimator, proposed by Newton and Raftery [1994], is perhaps the simplest of all estimators and is based on the following identity,

$$\frac{1}{p(\mathbf{y}|\boldsymbol{\theta})} = \frac{\int_z p(\mathbf{z}|\boldsymbol{\theta})d\mathbf{z}}{p(\mathbf{y}|\boldsymbol{\theta})} = \int_z \frac{p(\mathbf{z}|\mathbf{y},\boldsymbol{\theta})}{p(\mathbf{y}|\mathbf{z},\boldsymbol{\theta})}d\mathbf{z} \approx \frac{1}{S}\sum_{s=1}^{S}\left(\frac{1}{p(\mathbf{y}|\mathbf{z}^{(s)},\boldsymbol{\theta})}\right) \tag{2.4}$$

where $\mathbf{z}^{(s)}$ are samples from the posterior, obtained using a MCMC sampler. Although this estimate is consistent, it might be quite imprecise since the inverse likelihood does not always have finite variance [Chib, 1995].

Importance sampling (IS)

An alternative approach is based on importance sampling, where we sample from a proposal distribution $q(\mathbf{z})$ chosen such that it is "close" to the desired distribution but is easy to sample from. The marginal likelihood estimate can be obtained as shown below.

$$p(\mathbf{y}|\boldsymbol{\theta}) = \int_z p(\mathbf{y},\mathbf{z}|\boldsymbol{\theta})d\mathbf{z} = \int_z \frac{p(\mathbf{y},\mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})}q(\mathbf{z})d\mathbf{z} \approx \frac{1}{S}\sum_{s=1}^{S}\frac{p(\mathbf{y},\mathbf{z}^{(s)}|\boldsymbol{\theta})}{q(\mathbf{z}^{(s)})} \tag{2.5}$$

where $\mathbf{z}^{(s)}$ are samples from $q(\mathbf{z})$. Given that $q(\mathbf{z}) \neq 0$ whenever $p(\mathbf{y},\mathbf{z}|\boldsymbol{\theta}) \neq 0$, the above estimate converges to the true marginal likelihood for $S \to \infty$. The accuracy of the marginal likelihood estimate depends directly on the variability of the ratio $p(\mathbf{y},\mathbf{z}^{(s)}|\boldsymbol{\theta})/q(\mathbf{z}^{(s)})$, also known as importance weights. Better estimates are obtained when these weights have lower variances since, for large variances, the estimate will be based only on few points with large weights, giving rise to inaccurate estimates. The weights have low variance

if $q(\mathbf{z})$ is a close approximation of the joint distribution. However, when $\mathbf{z}$ is high-dimensional, finding good proposal distributions is difficult, limiting the applicability of this method.

Annealed importance sampling (AIS)

Annealed importance sampling (AIS) [Neal, 2001], a special case of larger family of Sequential Monte Carlo samplers [Moral et al., 2006], constructs a proposal distribution from a sequence of distributions $p_t(\mathbf{z})$ for $t = 0, 1, \ldots, T$. We pick distributions p_t such that we can sample from them, either by using Monte-Carlo or MCMC, and the distribution p_T is the target distribution. We should be able to evaluate a function $f_t(\mathbf{z})$ which is proportional to $p_t(\mathbf{z})$ and simulate some Markov chain transition, $\mathcal{T}_t$, that leaves p_t invariant. One useful construction in case of LGMs is the following,

$$f_t(\mathbf{z}) = p(\mathbf{y}|\mathbf{z}, \boldsymbol{\theta})^{\tau_t} \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{2.6}$$

where $1 = \tau_T > \tau_{T-1} > \ldots > \tau_1 = 0$. Sampling from f_0 is equivalent to sampling from the prior, while samples from f_T are from the desired posterior distribution. Methods described in the previous section can be used to draw samples from p_t and also to specify the Markov chain transition $\mathcal{T}_t$.

Given this sequence of distributions, we start sampling $\mathbf{z}_0$ from $p_0(\mathbf{z})$, then generate a sample $\mathbf{z}_1$ from $\mathcal{T}_1$, followed subsequently by sampling $\mathbf{z}_t$ from $\mathcal{T}_t$ for $t = 2, 3, \ldots, T-1$. This process gives us one sampled sequence $\mathbf{z}_0^{(s)}, \mathbf{z}_1^{(s)}, \ldots, \mathbf{z}_{T-1}^{(s)}$, which we denote by $\mathbf{Z}^{(s)}$. We repeat this process S times and compute an estimate of the marginal likelihood as shown below.

$$p(\mathbf{y}|\boldsymbol{\theta}) \approx \frac{1}{S} \sum_{s=1}^{S} \frac{f_{T-1}(\mathbf{z}_{T-1}^{(s)})}{f_T(\mathbf{z}_{T-1}^{(s)})} \frac{f_{T-2}(\mathbf{z}_{T-2}^{(s)})}{f_{T-1}(\mathbf{z}_{T-2}^{(s)})} \cdots \frac{f_1(\mathbf{z}_1^{(s)})}{f_2(\mathbf{z}_1^{(s)})} \frac{f_0(\mathbf{z}_0^{(s)})}{f_1(\mathbf{z}_0^{(s)})} \tag{2.7}$$

Just like IS, this estimate converges to the true marginal likelihood as $S \to \infty$. However, here we have a better control over the variability of importance weights than IS. For example, high variability can result from using transitions of each distribution that do not bring the distribution close to the equilibrium. This variability can be reduced by increasing the number of iterations of the MCMC sampler. Another sources of variability is the use of finite number of distributions between p_0 and p_T. This variability can be reduced by using a large number of distributions. Dimensionality of $\mathbf{z}$ also affects the variance, but the effect is less severe than in IS. Neal [2001] discusses a simple example where the performance of AIS degrades linearly

with the dimensionality of $\mathbf{z}$ as opposed to an exponential degradation for IS. See Neal [2001] for a discussion of how these factors affect performance of AIS. In summary, AIS can potentially lead to a low variance estimate of the marginal likelihood, but at a huge computational cost.

The Chib estimator

A simple method for estimating marginal likelihood using Gibbs sampling or MH algorithm is proposed by Chib [Chib, 1995; Chib and Jeliazkov, 2001]. The Chib estimator makes use of the following *basic marginal likelihood* identity to estimate the marginal likelihood.

$$p(\mathbf{y}|\boldsymbol{\theta}) = \frac{p(\mathbf{y}|\mathbf{z},\boldsymbol{\theta})p(\mathbf{z}|\boldsymbol{\theta})}{p(\mathbf{z}|\mathbf{y},\boldsymbol{\theta})} \approx \frac{p(\mathbf{y}|\mathbf{z}^*,\boldsymbol{\theta})p(\mathbf{z}^*|\boldsymbol{\theta})}{\hat{p}(\mathbf{z}^*|\mathbf{y},\boldsymbol{\theta})} \tag{2.8}$$

where $\mathbf{z}^*$ is a chosen value of $\mathbf{z}$ and $\hat{p}(\mathbf{z}^*|\mathbf{y},\boldsymbol{\theta})$ is an estimate of $p(\mathbf{z}|\mathbf{y},\boldsymbol{\theta})$ at $\mathbf{z}^*$. The above identity holds for all $\mathbf{z}^*$, but it is recommended to choose a high density point to obtain a good estimate of the denominator. In case of Gibbs sampling with auxiliary variables, Chib [1995] suggests an approach to estimate the denominator. Denoting the auxiliary variables by $\mathbf{u}$, the auxiliary-variable Gibbs sampler alternates between sampling from $p(\mathbf{z}|\mathbf{y},\mathbf{u},\boldsymbol{\theta})$ and $p(\mathbf{u}|\mathbf{y},\mathbf{z},\boldsymbol{\theta})$. The denominator then can be estimated from the samples of $\mathbf{u}$, using the following identity:

$$\hat{p}(\mathbf{z}^*|\mathbf{y},\boldsymbol{\theta}) = \int_u p(\mathbf{z}^*|\mathbf{y},\mathbf{u},\boldsymbol{\theta})p(\mathbf{u}|\mathbf{y},\boldsymbol{\theta})d\mathbf{u} \approx \frac{1}{S}\sum_{s=1}^{S} p(\mathbf{z}^*|\mathbf{y},\mathbf{u}^{(s)},\boldsymbol{\theta}) \tag{2.9}$$

since $\mathbf{u}^{(s)}$ is a sample from the marginal $p(\mathbf{u}|\mathbf{y},\boldsymbol{\theta})$, and the distribution $p(\mathbf{z}^*|\mathbf{y},\mathbf{u},\boldsymbol{\theta})$ is available up to the normalization constant, e.g. it is multivariate normal in case of binary probit link. The standard problems such as mixing of MCMC still affects this estimator since it is not easy to obtain good samples for estimation in Eq. 2.9. The Chib estimator also suffers from other problems when the model is unidentifiable; see Neal [1999] for details.

2.2.3 Parameter Estimation

MCMC methods can also be directly used to sample from the joint distribution $p(\mathbf{z}_1,\mathbf{z}_2,\ldots,\mathbf{z}_N,\boldsymbol{\theta}|\mathbf{Y})$, for example, using HMC [Mohamed et al., 2008], Gibbs sampling [Salakhutdinov and Mnih, 2008a], slice sampling [Murray and Adams, 2010], etc. Again, these methods suffer from similar problems

discussed for sampling from $p(\mathbf{z}|\mathbf{y}, \boldsymbol{\theta})$. In particular, they exhibit slow mixing due to a strong coupling between $\mathbf{z}_n$ and $\boldsymbol{\theta}$ [Murray and Adams, 2010].

An easier problem is to compute just a point estimate of $\boldsymbol{\theta}$, for which stochastic versions of expectation maximization (EM) algorithm can be used. The most popular version is the Monte Carlo expectation maximization (MCEM) [Wei and Tanner, 1990]. This approach uses samples from the posterior $p(\mathbf{z}_n|\mathbf{y}_n, \boldsymbol{\theta})$, obtained either using Monte Carlo or MCMC, to compute an approximation to the marginal likelihood. The approximation is constructed using a procedure similar to the EM algorithm. The EM algorithm is an iterative procedure where we obtain a lower bound to the marginal likelihood at the current parameter estimate, say $\boldsymbol{\theta}^t$, and maximize this lower bound to obtain the next estimate $\boldsymbol{\theta}^{t+1}$. To find a lower bound to the marginal likelihood, we consider the log of the marginal likelihood $\mathcal{L}(\boldsymbol{\theta})$ given in Eq. 2.10 and multiply and divide by the posterior distribution $p(\mathbf{z}_n|\mathbf{y}_n, \boldsymbol{\theta}^t)$.

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \log \int p(\mathbf{y}_n, \mathbf{z}_n|\boldsymbol{\theta}) d\mathbf{z}_n = \sum_{n=1}^{N} \log \int p(\mathbf{z}_n|\mathbf{y}_n, \boldsymbol{\theta}^t) \frac{p(\mathbf{y}_n, \mathbf{z}_n|\boldsymbol{\theta})}{p(\mathbf{z}_n|\mathbf{y}_n, \boldsymbol{\theta}^t)} d\mathbf{z}_n \tag{2.10}$$

As log is a concave function, we obtain a lower bound using Jensen's inequality, shown in Eq. 2.11 below, where we ignore the entropy term since it does not depend on $\boldsymbol{\theta}$. A Monte-Carlo approximation to this lower bound can be obtained using posterior samples as shown in Eq. 2.12.

$$\mathcal{L}(\boldsymbol{\theta}) \geq \mathcal{L}_J(\boldsymbol{\theta}|\boldsymbol{\theta}^t) := \sum_{n=1}^{N} \int_{z_n} p(\mathbf{z}_n|\mathbf{y}_n, \boldsymbol{\theta}^t) \log p(\mathbf{y}_n, \mathbf{z}_n|\boldsymbol{\theta}) d\mathbf{z}_n + \text{cnst} \tag{2.11}$$

$$\approx \mathcal{L}_{JS}(\boldsymbol{\theta}|\boldsymbol{\theta}^t) := \sum_{n=1}^{N} \frac{1}{S_t} \sum_{s=1}^{S_t} \log p(\mathbf{y}_n, \mathbf{z}_n^{(s)}|\boldsymbol{\theta}) + \text{cnst} \tag{2.12}$$

This step gives us an approximation of the sufficient statistics, using which we can obtain either closed-form updates for $\boldsymbol{\theta}$ or find approximate gradients to be used in the M-step [McCulloch, 1997].

One major implementation issue with MCEM is the specification of sample size S_t [Wei and Tanner, 1990]. As the algorithm approaches the maximum, the Monte Carlo approximation error becomes significant. So, for a constant S_t, the algorithm may reach somewhere close to the solution and then wander around it. Hence the algorithm may not converge. The solution is to increase S_t with time, however the rate at which S_t needs to be

increased usually depends on the problem, making an efficient implementation difficult [Booth and Hobert, 1999; Levine and Casella, 2001].

An important extension, called the stochastic approximation EM (SAEM), solves the convergence problem to some extent [Delyon et al., 1999]. In SAEM, the lower bound is obtained using a weighted combination of the current lower bound and the previous lower bound, i.e. the new lower bound is $\gamma_t \mathcal{L}_{JS}(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1}) + (1-\gamma_t)\mathcal{L}_{JS}(\boldsymbol{\theta}|\boldsymbol{\theta}_t)$ for some $\gamma_t > 0$. The advantage of this method is that it efficiently reuses the samples collected in previous iterations. This also makes implementation easier since we do not need to vary the sample size with t. The convergence is assured even for a constant S_t, provided that γ_t's are such that $\sum_t \gamma_t = \infty$ and $\sum_t \gamma_t^2 < \infty$. Another advantage is that the sequence of step size can be decided beforehand, e.g. Polyak and Juditsky [1992] show that for step size $\gamma_t \propto (1/t)^\alpha, 1/2 < \alpha < 1$, the algorithm converges at an optimum rate. However, in practice, convergence rate of the algorithm is very sensitive to the choice of step size. Large step sizes, e.g. α close to $1/2$, bring the algorithm quickly to the neighborhood of the solution, but inflate the Monte Carlo error, while small step sizes result in a fast reduction of Monte Carlo error, but slow down the convergence.

There are many other implementation issues with these algorithms, making their use cumbersome. For example, choice of sampling method is critical to their performance. Use of Monte Carlo methods is preferred since they generate independent samples, but they are highly inefficient in high dimensions [Booth and Hobert, 1999]. Samples from MCMC methods can be used but they are computationally intensive, since every EM iteration requires some burn-in followed by long runs of MCMC to make sure that MCMC has converged [Levine and Casella, 2001]. In addition, MCMC makes it difficult to assess the convergence of the EM algorithm.

2.3 Deterministic Methods

Deterministic methods require less computation time than MCMC, and perform better than the non-Bayesian approaches. The problem, however, is that they are not general enough and their applicability is limited.

2.3.1 Laplace's Method

Laplace's method approximates the integral of the form $\int \exp[f(\mathbf{z})]d\mathbf{z}$, where $f(\mathbf{z})$ is a function satisfying some regularity conditions. In this method, we first compute the mode $\mathbf{m}$ of $f(\mathbf{z})$ and its curvature $\mathbf{V}$ at the mode; both

quantities defined below.

$$\mathbf{m} := \arg\max_{z} f(\mathbf{z}) \tag{2.13}$$

$$\mathbf{V} := -\left[\frac{\partial^2 f(\mathbf{z})}{\partial \mathbf{z} \partial \mathbf{z}^T}\right]^{-1}_{\mathbf{z}=\mathbf{m}} \tag{2.14}$$

Then, we take a first-order Taylor series expansion of $f(\mathbf{z})$ around $\mathbf{m}$, as shown in Eq. 2.15. The second term in the expansion corresponds to a Gaussian, for which the normalizing constant is known, giving us the approximation in Eq. 2.16.

$$\int_z \exp[f(\mathbf{z})] d\mathbf{z} \approx \int_z \exp\left[f(\mathbf{m}) - \tfrac{1}{2}(\mathbf{z}-\mathbf{m})\mathbf{V}^{-1}(\mathbf{z}-\mathbf{m})\right] d\mathbf{z} \tag{2.15}$$

$$= e^{f(\mathbf{m})}(|2\pi\mathbf{V}|)^{1/2} \tag{2.16}$$

We now discuss the use of Laplace's method to accomplish our learning objectives. Tierney and Kadane [1986] were the first one to use Laplace's method for Bayesian inference. They define the function as follows,

$$f(\mathbf{z}_n) := \log p(\mathbf{y}_n, \mathbf{z}_n|\boldsymbol{\theta}) \tag{2.17}$$

We denote the mode and curvature by $\mathbf{m}_n$ and $\mathbf{V}_n$ respectively. For LGMs, computing $\mathbf{m}_n$ is easy since $f(\mathbf{z}_n)$ is a concave function, and $\mathbf{V}_n$ is usually available in closed form. These quantities can be used to approximate the posterior as follows,

$$p(\mathbf{z}_n|\mathbf{y}_n, \boldsymbol{\theta}) \propto p(\mathbf{y}_n, \mathbf{z}_n|\boldsymbol{\theta}) = e^{f(\mathbf{z}_n)} \tag{2.18}$$

$$\approx e^{f(\mathbf{m}_n)} \exp\left[-\tfrac{1}{2}(\mathbf{z}_n - \mathbf{m}_n)^T \mathbf{V}_n^{-1}(\mathbf{z}_n - \mathbf{m}_n)\right] \tag{2.19}$$

$$\propto \mathcal{N}(\mathbf{z}_n|\mathbf{m}_n, \mathbf{V}_n) \tag{2.20}$$

Marginal likelihood estimate is also straightforward,

$$\log p(\mathbf{y}_n|\boldsymbol{\theta}) = \log \int e^{f(\mathbf{z}_n)} d\mathbf{z}_n \approx f(\mathbf{m}_n) + \frac{1}{2}\log(|2\pi\mathbf{V}_n|) \tag{2.21}$$

$$= \frac{1}{2}\log(|2\pi\mathbf{V}_n|) + \log p(\mathbf{m}_n|\boldsymbol{\theta}) + \sum_{d=1}^{D} \log p(y_{dn}|\mathbf{m}_n, \boldsymbol{\theta}) \tag{2.22}$$

For parameter estimation, the above marginal likelihood approximation can be optimized with respect to $\boldsymbol{\theta}$.

Note the similarity between the non-Bayesian method and Laplace's approximation. If we ignore the first term, i.e. the determinant of $\mathbf{V}_n$, the

maximization with respect to $\boldsymbol{\theta}$ will be exactly equal to the non-Bayesian approach discussed in Section 2.1. Inclusion of $\mathbf{V}_n$, however, takes the uncertainty into account and solves the problems associated with non-Bayesian approaches to some extent.

A major drawback of Laplace's approximation is that it is based only on the mode of the posterior distribution, and hence fails to capture important global properties [Bishop, 2006]. For distributions, where mode is not a representative of the spread of the distribution, the covariance estimates are not accurate, giving rise to inaccurate marginal likelihood estimates. Kuss and Rasmussen [2005] discuss this issue for binary Gaussian process classification. In that case, the posterior distribution is highly skewed since the binary likelihoods essentially "chop" the Gaussian prior off. As a result of this, the mode remains close to the origin, even though the distribution contains a lot of mass away from the mode. The situation worsens as the dimensionality grows, since, for many Gaussian distributions, most of the mass is contained in a thin ellipsoid shell away from the mean [MacKay, 2003, Chapter 29.2].

2.3.2 Integrated Nested Laplace Approximation

A recent approach, called the integrated nested Laplace approximation (INLA), is proposed by Rue et al. [2009]. This approach aims to integrate out $\boldsymbol{\theta}$ numerically. They assume that dimension of $\boldsymbol{\theta}$ is small, around 6-7, making numerical integration possible. They consider the special case of LGMs where the covariance $\boldsymbol{\Sigma}$ is parameterized using $\boldsymbol{\theta}$ with one latent variable z_d for each y_d, similar to the Gaussian processes; see Fig. 1.2(c). Motivated by spatial statistics, they also assume sparse dependency among the latent variables z_d, although the method works for dense models as well. Their objective is to compute approximations to the marginals $p(z_d|\mathbf{y})$ using the following identity,

$$p(z_d|\mathbf{y}) = \int_{\theta} p(z_d|\mathbf{y}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta} \tag{2.23}$$

Rue et al. [2009] use Laplace's approximation to build approximations to $p(\boldsymbol{\theta}|\mathbf{y})$ and $p(z_d|\mathbf{y}, \boldsymbol{\theta})$, and then numerically integrate out $\boldsymbol{\theta}$ which is possible for small number of $\boldsymbol{\theta}$. The nested use of Laplace method along with the numerical integration gives the approach its name: the integrated "nested" Laplace approximation.

We now briefly describe the approach. First, we build an approximation to $p(\boldsymbol{\theta}|\mathbf{y})$. We do so in a "non-parametric" way using the identity given in

Eq. 2.24. The identity can be derived easily using the Bayes rule. Next, we substitute the Laplace approximation $\mathcal{N}(\mathbf{z}|\mathbf{m}(\boldsymbol{\theta}), \mathbf{V}(\boldsymbol{\theta}))$ to $p(\mathbf{z}|\boldsymbol{\theta}, \mathbf{y})$ and evaluate the ratio at $\mathbf{z} = \mathbf{m}(\boldsymbol{\theta})$ in Eq. 2.25 to get the final approximation in Eq. 2.26. Note that we denote $\mathbf{m}(\boldsymbol{\theta})$ and $\mathbf{V}(\boldsymbol{\theta})$ since these quantities need to be recomputed for every $\boldsymbol{\theta}$.

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{z}, \boldsymbol{\theta}|\mathbf{y})}{p(\mathbf{z}|\boldsymbol{\theta}, \mathbf{y})} \propto \frac{p(\mathbf{y}, \mathbf{z}, \boldsymbol{\theta})}{p(\mathbf{z}|\boldsymbol{\theta}, \mathbf{y})} \tag{2.24}$$

$$\tilde{p}(\boldsymbol{\theta}|\mathbf{y}) \approx \frac{p(\mathbf{y}, \mathbf{z}, \boldsymbol{\theta})}{\mathcal{N}(\mathbf{z}|\mathbf{m}(\boldsymbol{\theta}), \mathbf{V}(\boldsymbol{\theta})}\Big|_{\mathbf{z}=\mathbf{m}(\boldsymbol{\theta})} \tag{2.25}$$

$$= \frac{p(\mathbf{y}|\mathbf{m}(\boldsymbol{\theta}), \boldsymbol{\theta}) p(\mathbf{m}(\boldsymbol{\theta})|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{|2\pi \mathbf{V}(\boldsymbol{\theta})|^{-1/2}} \tag{2.26}$$

We would like to point out that this estimate is same as the marginal likelihood estimate using Laplace approximation shown in Eq. 2.22, with an addition of $p(\boldsymbol{\theta})$ term. This can be verified by talking log of Eq. 2.26 (also see comments by Papaspoliopoulos in the discussion of Rue et al. [2009]). This estimate has been shown to be of poor quality compared to other approximation methods [Kuss and Rasmussen, 2005], and will affect the quality of numerical integration in the next step.

The second step is to do a grid-search over several values of $\boldsymbol{\theta}$ to obtain "good" points where $p(\boldsymbol{\theta}|\mathbf{y})$ is expected to have "significant" probabilities. These values of $\boldsymbol{\theta}$ are used to do a numerical integration over $\boldsymbol{\theta}$. The final step is to use Laplace's method again to obtain posterior approximations to the marginal $p(z_d|\mathbf{y}, \boldsymbol{\theta})$ at the sampled values of $\boldsymbol{\theta}$. We proceed, as before, using the Bayes rule to obtain the first equality in Eq. 2.27. Then, we substitute the Laplace approximation of $p(\mathbf{z}_{-d}|z_d, \boldsymbol{\theta}, \mathbf{y})$, and finally evaluate the ratio at the mode $\mathbf{m}_d(z_d, \boldsymbol{\theta})$ to get the final approximation below. Here again, we denote $\mathbf{m}(z_d, \boldsymbol{\theta})$ and $\mathbf{V}(z_d, \boldsymbol{\theta})$ since these need to be recomputed for every z_d and $\boldsymbol{\theta}$ (Rue et al. [2009] make some more approximations to reduce this computation).

$$\tilde{p}(z_d|\boldsymbol{\theta}, \mathbf{y}) = \frac{p(\mathbf{y}, \mathbf{z}, \boldsymbol{\theta})}{p(\mathbf{z}_{-d}|z_d, \boldsymbol{\theta}, \mathbf{y})} \approx \frac{p(y_d|z_d) p(z_d|\mathbf{m}_d(z_d, \boldsymbol{\theta}), \boldsymbol{\theta})}{|2\pi \mathbf{V}_d(\boldsymbol{\theta})|^{-1/2}} \tag{2.27}$$

Finally, we obtain the marginal approximation for z_d using $\tilde{p}(z_d|\boldsymbol{\theta}, \mathbf{y})$ and $\tilde{p}(\boldsymbol{\theta}|\mathbf{y})$ as follows,

$$\tilde{p}(z_d|\mathbf{y}) = \sum_{k=1}^{K} \tilde{p}(z_d|\boldsymbol{\theta}^{(k)}, \mathbf{y}) \tilde{p}(\boldsymbol{\theta}^{(k)}|\mathbf{y}) \Delta_k \tag{2.28}$$

where $\boldsymbol{\theta}^{(s)}$ are values obtained with the grid-search and Δ_k are the area weights in the space of $\boldsymbol{\theta}$.

In Rue et al. [2009], the authors demonstrate experimentally that the approximation is as accurate as MCMC methods but takes much less time. This has been supported by many other researcher (for details see discussions that follow the paper). The use of Laplace's method is not necessary for this method, and several new extensions have been proposed [Cseke and Heskes, 2010, 2011]. The main disadvantage of the approach is that it applies only to cases where the number of parameters is very small. Hence, the method does not generalize to many other LGMs, such as the factor model. Another issue is that they consider computation of posterior marginals, rather than the full posterior which may be useful in many applications.

2.3.3 Expectation Propagation

Another approximation method, called expectation-propagation (EP) [Minka, 2001], has been applied extensively to LGMs [Kuss and Rasmussen, 2005; Nickisch and Rasmussen, 2008; Seeger and Nickisch, 2011; Zoeter et al., 2005]. EP approximates the posterior distribution by maintaining expectations and iterating until these expectations are consistent for all variables.

We now briefly describe this procedure for the case when each dimension y_d depends only on z_d. We assume that the prior mean $\boldsymbol{\mu}$ is set to 0. EP computes a Gaussian approximation to the posterior by replacing discrete-data likelihoods by unnormalized Gaussian likelihoods, called the site functions, defined as $t(z_d, \overline{m}_d, \overline{v}_d, Z_d) = Z_d \mathcal{N}(z_d|\overline{m}_d, \overline{v}_d)$. This replacement gives us a posterior approximation as shown below.

$$p(\mathbf{z}|\mathbf{y}, \boldsymbol{\theta}) = \frac{p(\mathbf{z}|\boldsymbol{\theta})}{p(\mathbf{y}|\boldsymbol{\theta})} \prod_{d=1}^{D} p(y_d|z_d) \approx \frac{p(\mathbf{z}|\boldsymbol{\theta})}{p(\mathbf{y}|\boldsymbol{\theta})} \prod_{d=1}^{D} t(z_d, \overline{m}_d, \overline{v}_d, Z_d) \tag{2.29}$$

Completing the squares, we can express the approximation as a Gaussian distribution, $q(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{m}, \mathbf{V})$, with mean $\mathbf{m} = \mathbf{V}\text{diag}(\overline{\mathbf{v}})^{-1}\overline{\mathbf{m}}$ and covariance $\mathbf{V} = [\boldsymbol{\Sigma}^{-1} + \text{diag}(\overline{\mathbf{v}})]^{-1}$ where $\overline{\mathbf{v}}$ and $\overline{\mathbf{m}}$ are the vectors containing the site function parameters $\overline{v}_d$ and $\overline{m}_d$.

We need to set the site parameters appropriately to get a good approximation to the posterior. In EP, we do this iteratively visiting each site function and adjusting the site parameters to match moments under the approximate posterior distribution. To be precise, we first find the approximate marginal posterior $q_d(z_d)$ for d'th site by marginalizing out other latent variables, denoted by $\mathbf{z}_{-d}$. This is easy since the approximate distribution

is Gaussian, giving us the marginal shown below.

$$q_d(z_d) = \int q(\mathbf{z}) d\mathbf{z}_{-d} = \mathcal{N}(z_d|\widetilde{m}_i, \widetilde{v}_i) \tag{2.30}$$

with $\widetilde{v}_d = 1/(1/V_{dd} - 1/\overline{v}_d)$ and $\widetilde{m}_d = \widetilde{v}_d(m_d/V_{dd} - \overline{m}_d/\overline{v}_d)$. Given this marginal, we set the site parameters such that the moments of a site function under $q_d(z_d)$ are equal to the moments of the true likelihood, as shown below.

$$\int z_d^k p(y_d|z_d, \boldsymbol{\theta}) q_d(z_d) dz_d = \int z_d^k t(z_d, \overline{m}_d, \overline{v}_d, Z_d) q_d(z_d) dz_d \tag{2.31}$$

For Gaussian distribution, we need to solve this equation for $k = 0, 1$ and 2. Right hand side of above equation are simply the moments of a Gaussian and are available in closed form. The difficulty of obtaining left hand side depends on the form of the likelihood.

Computing the integral in the left hand side can be easy sometimes; e.g. for the probit likelihood, it is available in closed form. There are cases, however, for which it could be difficult. Seeger and Jordan [2004] discuss the difficulty with the multinomial-logit likelihood in the context of multi-class GP classification, stating that the EP approach is "fundamentally limited by the requirement of an efficient numerical integration in K dimensions", K being the number of categories [Seeger and Jordan, 2004, §4.3.1].

Another issue with EP is that the iterative moment matching procedure is not always guaranteed to converge and is known to have numerical problems for some likelihoods [Jylänki et al., 2011], for example, the site variances might be negative at times. It is possible to build provably convergent EP algorithm for some cases, e.g. see [Hernández-Lobato and Hernández-Lobato, 2011; Seeger and Nickisch, 2011]. However, designing such convergent generic EP algorithm remains an open research problem.

The site functions can be used to compute an approximation to the marginal likelihood since they are unnormalized Gaussian distributions, as shown below.

$$\log p(\mathbf{y}|\theta) \approx \log \int p(\mathbf{z}|\boldsymbol{\theta}) \prod_{d=1}^{D} t(z_d, \overline{m}_d, \overline{v}_d, Z_d) d\mathbf{z} \tag{2.32}$$

Similar to Laplace's method, the above approximation can be maximized to get an estimate for $\boldsymbol{\theta}$. In practice, it may not always be easy to compute this approximation, for example, in multi-class GPC [Seeger and Jordan, 2004, §5]. In such situations, the use of a lower bound based on Kullback-Leibler (KL) divergence is suggested. However, this leads to a non-standard and

usually non-monotonic optimization since the inference and learning steps do not optimize the same lower bound. These lower bounds are usually not convex, which further adds to the difficulty. Such a hybrid EM-EP procedure is used by Rattray et al. [2009], who also discuss the difficulty in computing EP approximations for parameter learning. Also, see Stern et al. [2009].

2.4 Summary

Non-conjugacy makes learning in discrete-data LGMs difficult. In this chapter, we reviewed many methods for learning and discussed their advantages and disadvantages. In summary,

1. Non-Bayesian approaches are fast, but do not always perform well due to overfitting and sensitivity to the regularization parameter.

2. MCMC methods have potential to perform well, but they suffer from slow mixing and require expert level knowledge for tuning and convergence diagnostics.

3. Deterministic methods provide a good alternative to MCMC and non-Bayesian approaches, since they could be as fast as non-Bayesian approaches, and at times, are as accurate as MCMC. However, deterministic methods discussed in this chapter are not general enough to achieve our learning objectives. For example, INLA works only for small number of parameters, while EP is troublesome for parameter estimation and suffers from convergence issues.

Chapter 3

Variational Learning of Discrete-Data LGMs

We reviewed many methods for learning LGMs in the previous chapter and showed that none of those methods provide satisfactory solutions to our problems. In this chapter, we introduce a variational approach based on *Evidence Lower BOund* (ELBO). The main advantage of this approach is that the lower bound can be used as a common objective function for all of our learning objectives. Unlike other deterministic methods, this approach does not suffer from convergence issues and is applicable to general settings, such as the parameter learning in factor models.

We start this chapter by introducing the variational approach based on ELBO optimization. We discuss two main challenges associated with the approach. First challenge is that the lower bound is not always tractable. We solve this problem by using local variational bounds (LVBs) to the intractable terms in ELBO. Second challenge is related to the computational inefficiencies associated with the optimization of the lower bound. We make three contributions in this regard. First, we show that the lower bound has useful concavity properties, making the optimization efficient. Second, we derive generalized gradient expressions for optimization. Third, we propose a fast convergent coordinate-ascent inference algorithm. In rest of the thesis, we will refer to the lower bound approach as the variational method, although there exist many other varieties of variational methods.

3.1 A Variational Approach Based on the Evidence Lower Bound

The marginal likelihood for discrete-data LGMs is intractable since the discrete-data likelihood is not conjugate to the Gaussian prior. We can, however, obtain a lower bound to the marginal likelihood using the Jensen inequality. This lower bound is called by many names, such as the evidence lower bound [Braun and McAuliffe, 2010], the Gaussian variational bound

[Challis and Barber, 2011], the Kullback-Leibler (KL) bound [Kuss and Rasmussen, 2005], or simply the Jensen lower bound [Jaakkola, 2001; Jordan et al., 1998; Wainwright and Jordan, 2008]. We will refer to this lower bound as the evidence lower bound, because other names might be confusing later in our context.

The variational method based on ELBO is similar in spirit to the expectation maximization (EM) algorithm, but is more general since it applies even to the models with intractable posterior distributions. The basic idea behind ELBO is to restrict the form of the posterior distribution to a tractable class of distributions. For LGMs, the Gaussian distribution is a suitable choice since the posterior distribution is very close to the Gaussian distribution due to the Gaussian prior. We denote the approximation to $p(\mathbf{z}_n|\mathbf{y}_n, \boldsymbol{\theta})$ by $q(\mathbf{z}_n|\boldsymbol{\gamma}_n) = \mathcal{N}(\mathbf{z}_n|\mathbf{m}_n, \mathbf{V}_n)$, where the set of *variational parameters* is denoted by $\boldsymbol{\gamma}_n = \{\mathbf{m}_n, \mathbf{V}_n\}$. We denote the set of $\boldsymbol{\gamma}_n$ by $\boldsymbol{\gamma}$.

To obtain the evidence lower bound to the marginal likelihood, we begin with log of the marginal likelihood $\mathcal{L}(\boldsymbol{\theta})$ shown in Eq. 3.1, and multiply and divide by $q(\mathbf{z}_n|\boldsymbol{\gamma}_n)$ as shown in Eq. 3.2.

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \log \int p(\mathbf{z}_n|\boldsymbol{\theta}) p(\mathbf{y}_n|\mathbf{z}_n, \boldsymbol{\theta}) d\mathbf{z}_n \tag{3.1}$$

$$= \sum_{n=1}^{N} \log \int q(\mathbf{z}_n|\boldsymbol{\gamma}_n) \frac{p(\mathbf{z}_n|\boldsymbol{\theta}) p(\mathbf{y}_n|\mathbf{z}_n, \boldsymbol{\theta})}{q(\mathbf{z}_n|\boldsymbol{\gamma}_n)} d\mathbf{z}_n \tag{3.2}$$

Since log is a concave function, we can use the Jensen inequality to obtain a lower bound to each of the terms under the summation, as shown below.

$$\underline{\mathcal{L}}_n^J(\boldsymbol{\theta}, \boldsymbol{\gamma}_n) := \int q(\mathbf{z}_n|\boldsymbol{\gamma}_n) \log \frac{p(\mathbf{z}_n|\boldsymbol{\theta})}{q(\mathbf{z}_n|\boldsymbol{\gamma}_n)} d\mathbf{z}_n + \int q(\mathbf{z}_n|\boldsymbol{\gamma}_n) \log p(\mathbf{y}_n|\mathbf{z}_n, \boldsymbol{\theta}) d\mathbf{z}_n \tag{3.3}$$

The first integral here is simply negative of the Kullback−Leibler (KL) divergence from the Gaussian posterior $q(\mathbf{z}_n|\boldsymbol{\gamma}_n) = \mathcal{N}(\mathbf{z}_n|\mathbf{m}_n, \mathbf{V}_n)$ to the Gaussian prior distribution $p(\mathbf{z}_n|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, and has a closed-form expression shown below (recall that L is the length of $\mathbf{z}_n$),

$$\log |\mathbf{V}_n \boldsymbol{\Sigma}^{-1}| - \mathrm{Tr}(\mathbf{V}_n \boldsymbol{\Sigma}^{-1}) - (\mathbf{m}_n - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{m}_n - \boldsymbol{\mu}) + L \tag{3.4}$$

The second integral in Eq. 3.3 does not always have a tractable expression, but can be simplified. Recall from the definition of LGM that the likelihood is expressed in terms of the predictor $\boldsymbol{\eta}_{dn} = \mathbf{W}_d^T \mathbf{z}_n + \mathbf{w}_{0d}$. To be

precise, $p(\mathbf{y}_n|\mathbf{z}_n, \boldsymbol{\theta}) = \prod_d p(y_{dn}|\boldsymbol{\eta}_{dn}, \boldsymbol{\theta})$. To simplify the integral, we apply a change of variable to express the integral in terms of $\boldsymbol{\eta}_{dn}$ instead of $\mathbf{z}_n$. Since $\boldsymbol{\eta}_{dn}$ is a linear function of $\mathbf{z}_n$, we can easily derive its distribution. The distribution is given as follows: $q(\boldsymbol{\eta}_{dn}|\widetilde{\boldsymbol{\gamma}}_{dn}) := \mathcal{N}(\boldsymbol{\eta}_{dn}|\widetilde{\mathbf{m}}_{dn}, \widetilde{\mathbf{V}}_{dn})$ with

$$\widetilde{\mathbf{m}}_{dn} := \mathbf{W}_d^T \mathbf{m}_n + \mathbf{w}_{0d} \qquad \widetilde{\mathbf{V}}_{dn} := \mathbf{W}_d^T \mathbf{V}_n \mathbf{W}_d \tag{3.5}$$

Substituting this, we get a simplified expression for the second integral,

$$\int q(\mathbf{z}_n|\boldsymbol{\gamma}_n) \log p(\mathbf{y}_n|\mathbf{z}_n, \boldsymbol{\theta}) d\mathbf{z}_n = \sum_{d=1}^{D} \mathbb{E}_{q(\eta_{dn}|\widetilde{\gamma}_{dn})}[\log p(y_{dn}|\boldsymbol{\eta}_{dn}, \boldsymbol{\theta})] \tag{3.6}$$

We substitute Eq. 3.4 and 3.6 in the lower bound of Eq. 3.3 to get the evidence lower bound defined below.

Definition 3.1.1. *For the LGM defined in Section 1.1, the marginal likelihood $\mathcal{L}(\boldsymbol{\theta})$ is lower bounded by the evidence lower bound defined below,*

$$\underline{\mathcal{L}}^J(\boldsymbol{\theta}, \boldsymbol{\gamma}) := \sum_{n=1}^{N} \underline{\mathcal{L}}_n^J(\boldsymbol{\theta}, \boldsymbol{\gamma}_n) \tag{3.7}$$

where $\boldsymbol{\gamma}_n = \{\mathbf{m}_n, \mathbf{V}_n\}$ is the set of mean and covariance of approximate Gaussian posterior $q(\mathbf{z}_n|\boldsymbol{\gamma}_n)$, $\boldsymbol{\gamma}$ is the set of $\boldsymbol{\gamma}_n$, and $\underline{\mathcal{L}}_n^J$ is the lower bound to the marginal likelihood $\log p(\mathbf{y}_n|\boldsymbol{\theta})$ of n'th data vector and is defined below.

$$\begin{aligned}\underline{\mathcal{L}}_n^J(\boldsymbol{\theta}, \boldsymbol{\gamma}_n) := \tfrac{1}{2}\left[\log|\mathbf{V}_n\boldsymbol{\Sigma}^{-1}| - Tr(\mathbf{V}_n\boldsymbol{\Sigma}^{-1}) - (\mathbf{m}_n - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{m}_n - \boldsymbol{\mu}) + L\right] \\ + \sum_{d=1}^{D} \mathbb{E}_{q(\eta_{dn}|\widetilde{\gamma}_{dn})}[\log p(y_{dn}|\boldsymbol{\eta}_{dn}, \boldsymbol{\theta})]\end{aligned} \tag{3.8}$$

Recall that $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and covariance of the Gaussian prior, L is the length of $\mathbf{z}_n$, $p(y_{dn}|\boldsymbol{\eta}_{dn}, \boldsymbol{\theta})$ is the likelihood of the discrete-data y_{dn} with $\boldsymbol{\eta}_{dn}$ as the linear predictor, and the approximate distribution $q(\boldsymbol{\eta}_{dn}|\widetilde{\boldsymbol{\gamma}}_{dn})$ is defined in Eq. 3.5.

The first line in Eq. 3.8 is the negative of the KL divergence from the Gaussian posterior $q(\mathbf{z}|\boldsymbol{\gamma}_n)$ to the Gaussian prior distribution $p(\mathbf{z}|\boldsymbol{\theta})$. The second line is the expectation of the log-likelihood under the approximate posterior of the predictor $\boldsymbol{\eta}_{dn}$. Intuitively, the two terms "push" the posterior in opposite directions. The KL divergence term keeps the posterior close to the prior, while the second term brings it close to the data by increasing the expected likelihood of the data.

The variational approach based on ELBO is related to many other existing variational methods. For example, posterior inference using ELBO is exactly same as the popular variational inference based on the minimization of the KL divergence between the approximate posterior $q(\mathbf{z}_n|\boldsymbol{\gamma}_n)$ and the true posterior $p(\mathbf{z}_n|\mathbf{y}_n, \boldsymbol{\theta})$ [Jaakkola, 2001; Jordan et al., 1998; Wainwright and Jordan, 2008]. Also, see variational EM algorithm of Buntine [2002] which shows both the ELBO and the KL divergence (Eq. 5 and 6 in the paper). Similarly, the *mean field* method is same as the ELBO approach, but puts an additional factorization assumption on $q(\mathbf{z}|\boldsymbol{\gamma})$ [Knowles and Minka, 2011; Paisley et al., 2012].

3.2 Intractability of ELBO

The tractability of the evidence lower bound of Eq. 3.8 depends on the tractability of the expectation term $\mathbb{E}_{q(\eta_{dn}|\widetilde{\gamma}_{dn})}[\log p(y_{dn}|\boldsymbol{\eta}_{dn}, \boldsymbol{\theta})]$. For some likelihoods this term is tractable, while for others it is not. We first give a few examples of likelihoods which give rise to tractable ELBO, and then discuss the difficulty in discrete-data likelihoods.

The Gaussian likelihood is given as follows: $p(y_{dn}|\eta_{dn}, \psi_d) = \mathcal{N}(y_{dn}|\eta_{dn}, \psi_d)$ for real y_{dn}. This likelihood gives rise to the simplest and most widely used LGMs such as probabilistic PCA and factor analysis; see Section 1.2.4 for details. The lower bound for these *Gaussian LGMs* is tractable, since the expectation term is available in closed form as shown below,

$$\begin{aligned}
&\mathbb{E}_{q(\eta_{dn}|\widetilde{\gamma}_{dn})}[\log p(y_{dn}|\eta_{dn}, \psi_d)] && (3.9)\\
&= \mathbb{E}_{q(\eta_{dn}|\widetilde{\gamma}_{dn})}[\log \mathcal{N}(y_{dn}|\eta_{dn}, \psi_d)] && (3.10)\\
&= \mathbb{E}_{q(\eta_{dn}|\widetilde{\gamma}_{dn})}[-\tfrac{1}{2}\log(2\pi\psi_d) - \tfrac{1}{2}(y_{dn} - \eta_{dn})^2/\psi_d] && (3.11)\\
&= -\tfrac{1}{2}\log(2\pi\psi_d) - \tfrac{1}{2}[(y_{dn} - \widetilde{m}_{dn})^2 + \widetilde{v}_{dn}]/\psi_d && (3.12)
\end{aligned}$$

The Poisson likelihood is defined as $p(y_{dn}|\eta_{dn}) = \exp(y_{dn}\eta_{dn} - e^{\eta_{dn}})/y_{dn}!$ for a non-negative integer y_{dn}. The expectation term can be written in closed form as shown below,

$$\begin{aligned}
\mathbb{E}_{q(\eta|\widetilde{\gamma}_{dn})}[\log p(y_{dn}|\eta_{dn})] &= \mathbb{E}_{q(\eta|\widetilde{\gamma}_{dn})}[y_{dn}\eta_{dn} - e^{\eta_{dn}} - \log y_{dn}!] && (3.13)\\
&= y_{dn}\widetilde{m}_{dn} - e^{\widetilde{m}_{dn} + \widetilde{v}_{dn}/2} - \log y_{dn}! && (3.14)
\end{aligned}$$

where we use the identity given in Appendix A.1 to get the second term.

The stochastic volatility model uses the following likelihood: $p(y_{dn}|\eta_{dn}) = \mathcal{N}(0, e^{\eta_{dn}})$ for real y_{dn} [Rue et al., 2009]. Again, the expectation term is available in closed form as we show below,

$$\mathbb{E}_{q(\eta_{dn}|\widetilde{\gamma}_{dn})}[\log \mathcal{N}(y_{dn}|0, e^{\eta_{dn}})] \tag{3.15}$$
$$= \mathbb{E}_{q(\eta_{dn}|\widetilde{\gamma}_{dn})}[-\tfrac{1}{2}\log(2\pi e^{\eta_{dn}}) - \tfrac{1}{2}y_{dn}^2/e^{\eta_{dn}}] \tag{3.16}$$
$$= -\tfrac{1}{2}\log(2\pi) - \tfrac{1}{2}\widetilde{m}_{dn} - \tfrac{1}{2}y_{dn}^2 e^{-\widetilde{m}_{dn}+\widetilde{v}_{dn}/2} \tag{3.17}$$

where we use the identity given in Appendix A.1 to get the third term.

Intractable discrete-data likelihoods: Unfortunately, for most of the discrete-data likelihoods discussed in Section 1.3, the expectation term is not available in closed form. For example, for the Bernoulli logit likelihood $p(y_{dn} = 1|\eta_{dn}) = e^{\eta_{dn}}/(1 + e^{\eta_{dn}})$, the expectation shown below,

$$\mathbb{E}_{q(\eta_{dn}|\widetilde{\gamma}_{dn})}[\log p(y_{dn} = 1|\eta_{dn})] = \mathbb{E}_{q(\eta_{dn}|\widetilde{\gamma}_{dn})}[\eta_{dn} - \log(1 + e^{\eta_{dn}})] \tag{3.18}$$

is intractable due to the $\log(1 + e^{\eta_{dn}})$ term. Similarly, for the multinomial logit likelihood $p(y_{dn} = k|\boldsymbol{\eta}_{dn}) = e^{\eta_{kdn}}/\sum_j e^{\eta_{jdn}}$, the expectation shown below is intractable due to the log-sum-exp term.

$$\mathbb{E}_{q(\eta_{dn}|\widetilde{\gamma}_{dn})}[\log p(y_{dn} = k|\boldsymbol{\eta}_{dn})] = \mathbb{E}_{q(\eta_{dn}|\widetilde{\gamma}_{dn})}[\eta_{kdn} - \log \sum_j e^{\eta_{jdn}}] \tag{3.19}$$

3.3 Tractable ELBOs Using Local Variational Bounds (LVBs)

As discussed in the previous section, the expectation term is intractable for many discrete-data likelihoods. In this thesis, we use local variational bounds to make this term tractable, i.e. we design functions $\underline{f}$ such that they lower bound the expectation of the log-likelihood, as shown below.

$$\mathbb{E}_{q(\eta|\widetilde{\gamma})}[\log p(y|\boldsymbol{\eta})] \geq \underline{f}(y, \widetilde{\boldsymbol{\gamma}}, \boldsymbol{\alpha}) \tag{3.20}$$

where $\boldsymbol{\alpha}$ are the *local variational parameters*, which are optimized to get the tightest lower bound. The function $\underline{f}$ might not always have local variational parameters, for example, in case of Poisson distribution in Eq. 3.13, there are no local parameters.

Substituting the LVB in Eq. 3.8, we get a tractable lower bound below.

$$\underline{\mathcal{L}}_n(\boldsymbol{\theta}, \boldsymbol{\gamma}_n, \boldsymbol{\alpha}_n) := \tfrac{1}{2}\left[\log|\mathbf{V}_n\boldsymbol{\Sigma}^{-1}| - \mathrm{Tr}(\mathbf{V}_n\boldsymbol{\Sigma}^{-1}) - (\mathbf{m}_n - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{m}_n - \boldsymbol{\mu}) + L\right] + \sum_{d=1}^{D} \underline{f}(y_{dn}, \widetilde{\boldsymbol{\gamma}}_{dn}, \boldsymbol{\alpha}_{dn}) \tag{3.21}$$

Here, the local variational parameter $\boldsymbol{\alpha}_{dn}$ depends on d and n since we need to optimize LVB for each y_{dn}. We also get the following new lower bound on the marginal likelihood,

$$\mathcal{L}(\boldsymbol{\theta}) \geq \underline{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\alpha}) := \sum_{n=1}^{N} \underline{\mathcal{L}}_n(\boldsymbol{\theta}, \boldsymbol{\gamma}_n, \boldsymbol{\alpha}_n) \tag{3.22}$$

where we denote the set of all $\boldsymbol{\alpha}_{dn}$ by $\boldsymbol{\alpha}_n$ and set of all $\boldsymbol{\alpha}_n$ by $\boldsymbol{\alpha}$.

It is not necessary for $\underline{f}$ to have an analytical form, but we should be able evaluate the bound to be able to optimize ELBO. The function $\underline{f}$ could be an approximation as well, e.g. see Ahmed and Xing [2007]; Braun and McAuliffe [2010], and more recent results shown in Paisley et al. [2012]. Since a lower bound maintains the lower bounding property of ELBO, we focus mainly on the lower bounds rather than the approximations. We devote Chapter 4 and 5 entirely to discuss LVBs for binary and categorical data, respectively. We also briefly discuss the use of these LVBs to get bounds for ordinal data in Chapter 6.

3.4 Concavity of the Evidence Lower Bound

From this section onward, we focus on the computational aspects of ELBO. We start by discussing the concavity of ELBO. The following theorem, proved in Appendix A.2, establishes the concavity of ELBO.

Theorem 3.4.1. *Assuming that the LVB $\underline{f}(y_{dn}, \widetilde{\boldsymbol{\gamma}}_{dn}, \boldsymbol{\alpha}_{dn})$ is jointly concave with respect to $\widetilde{\boldsymbol{\gamma}}_{dn} = \{\widetilde{\mathbf{m}}_{dn}, \widetilde{\mathbf{V}}_{dn}\}$ and $\boldsymbol{\alpha}_{dn}$, we have the following:*

- *The lower bound $\underline{\mathcal{L}}_n(\boldsymbol{\theta}, \boldsymbol{\gamma}_n, \boldsymbol{\alpha}_n)$ of Eq. 3.21 is **strictly** jointly concave with respect to $\boldsymbol{\gamma}_n = \{\mathbf{m}_n, \mathbf{V}_n\}$ and $\boldsymbol{\alpha}_n$, given $\boldsymbol{\theta}$.*
- *The lower bound $\underline{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\alpha})$ of Eq. 3.22 is concave with respect to each of the following, $\boldsymbol{\mu}, \boldsymbol{\Sigma}^{-1}, \mathbf{W}$ and $\mathbf{w}_0$, for fixed $\boldsymbol{\gamma}$ and $\boldsymbol{\alpha}$. The lower bound is **strictly** concave with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}^{-1}$.*

Note that a function is (strictly) concave iff its Hessian is negative (semi)-definite [Boyd and Vandenberghe, 2004].

We conjecture that, for the second result in the theorem, the lower bound is jointly concave with respect to all the variables. This can be checked easily for the 1-D case, although proving a general case requires more effort. Also, the second result can be used to find out whether the lower bound is concave with respect to $\boldsymbol{\theta}$. For example, for Gaussian process, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are expressed in terms of $\boldsymbol{\theta}$ using the mean and covariance functions. The concavity with respect to $\boldsymbol{\theta}$ there depends on the mean and covariance functions, and can easily be determined using the chain rule.

Concavity results similar to ours have been discussed by Braun and McAuliffe [2010] for learning the discrete-choice models, and more recently by Challis and Barber [2011] for inference in Bayesian linear models.

We would like to point out the well known form of the lower bound of Eq. 3.21. Given $\mathbf{V}_n$, the function with respect to $\mathbf{m}_n$ is the nonlinear least square function. Similarly, given $\mathbf{m}_n$, the function with respect to $\mathbf{V}_n$ is similar to the graphical lasso [Friedman et al., 2008] or covariance selection problem [Dempster, 1972], with the difference that the argument is a covariance matrix instead of a precision matrix. These two objective functions are coupled through the non-linear term $\underline{f}$. Usually this term arises due to the prior distribution and may be non-smooth, for example, in graphical lasso. In our case, this term arises from the bound on the log-likelihood, and is smooth and usually jointly concave over $\mathbf{m}_n$ and $\mathbf{V}_n$.

The concavity of the lower bound and its similarity to many concave optimization problems has huge implications for the computational efficiency of variational learning. There is a vast literature on the efficient optimization of concave functions such as the least square and covariance selection problems. We can exploit this literature to design efficient algorithms for variational learning. We will present one such example in Section 3.6, where we design a fast convergent algorithm for inference in LGMs, such as Gaussian processes.

3.5 Variational Learning using Gradient Methods

We now describe how to perform different tasks of interest using ELBO. To obtain an approximation to the posterior $p(\mathbf{y}_n|\mathbf{z}_n, \boldsymbol{\theta})$, we maximize the ELBO of Eq. 3.21 with respect to $\boldsymbol{\alpha}_n$ and $\boldsymbol{\gamma}_n$ as shown below.

$$\max_{\boldsymbol{\gamma}_n, \boldsymbol{\alpha}_n} \underline{\mathcal{L}}_n(\boldsymbol{\theta}, \boldsymbol{\gamma}_n, \boldsymbol{\alpha}_n) \tag{3.23}$$

This optimization can be done easily by alternate maximization with respect to $\boldsymbol{\gamma}_n$ and $\boldsymbol{\alpha}_n$, both optimization involving concave function for most LVBs. For many LVBs, it is also possible to do an elimination of $\boldsymbol{\alpha}_n$ completely, followed by a maximization with respect to $\boldsymbol{\gamma}_n$. The approximate posterior is given by a Gaussian distribution $\mathcal{N}(\mathbf{z}_n|\mathbf{m}_n, \mathbf{V}_n)$. Plugging in the value of $\boldsymbol{\gamma}_n$ and $\boldsymbol{\alpha}_n$ at the maximum gives us a lower bound of the marginal likelihood, which can be used as the marginal likelihood estimate.

For parameter estimation, we iterate between optimizing with respect to $\boldsymbol{\gamma}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\theta}$ iteratively. Given a current estimate $\boldsymbol{\theta}^t$ at iteration t, we maximize $\underline{\mathcal{L}}_n$ to obtain $\boldsymbol{\gamma}_n^t$ and $\boldsymbol{\alpha}_n^t$ for all n, as shown in Eq. 3.24. Next, we optimize ELBO with respect to $\boldsymbol{\theta}$ by plugging in $\boldsymbol{\gamma}_n^t$ and $\boldsymbol{\alpha}_n^t$ as shown in Eq. 3.25.

$$\{\boldsymbol{\gamma}_n^t, \boldsymbol{\alpha}_n^t\} = \arg\max_{\boldsymbol{\gamma}_n, \boldsymbol{\alpha}_n} \underline{\mathcal{L}}_n(\boldsymbol{\theta}^t, \boldsymbol{\gamma}_n, \boldsymbol{\alpha}_n), \forall n \tag{3.24}$$

$$\boldsymbol{\theta}_n^t = \arg\max_{\boldsymbol{\theta}} \sum_{n=1}^{N} \underline{\mathcal{L}}_n(\boldsymbol{\theta}, \boldsymbol{\gamma}_n^t, \boldsymbol{\alpha}_n^t) \tag{3.25}$$

As shown before, both of these steps involve concave optimization, for which we can use any gradient based method such as LBFGS.

Finally, for prediction of a missing entry y_i^m given observed data vector $\mathbf{y}^o$ and $\boldsymbol{\theta}$, we first find the posterior approximation to $p(\mathbf{z}|\mathbf{y}^o, \boldsymbol{\theta})$, denote by $\mathcal{N}(\mathbf{z}|\mathbf{m}, \mathbf{V})$, and compute the predictive probability as shown below.

$$p(y_i^m|\mathbf{y}^o, \boldsymbol{\theta}) = \int_z p(y_i^m|\boldsymbol{\eta}) p(\mathbf{z}|\mathbf{y}^o, \boldsymbol{\theta}) d\mathbf{z} \approx \int_z p(y_i^m|\boldsymbol{\eta}) \mathcal{N}(\mathbf{z}|\mathbf{m}, \mathbf{V}) d\mathbf{z} \tag{3.26}$$

The above integral can be estimated using Monte-Carlo sampling, which is efficient since the posterior on $\mathbf{z}$ is a Gaussian distribution.

3.5.1 Generalized Gradient Expressions

All of the above optimization involve concave functions, for which gradient based methods can be used. We now give general expressions for gradients with respect to $\boldsymbol{\gamma}_n$ and $\boldsymbol{\theta}$. These gradients can be written in a general form in terms of the gradients of LVB $\underline{f}(y_{dn}, \widetilde{\boldsymbol{\gamma}}_{dn}, \boldsymbol{\alpha}_{dn})$, making it easy to implement the optimization routines. These gradients are defined below,

$$\mathbf{g}_{dn}^m := \frac{\partial \underline{f}_{dn}}{\partial \widetilde{\mathbf{m}}_{dn}} \quad , \quad \mathbf{G}_{dn}^v := \frac{\partial \underline{f}_{dn}}{\partial \widetilde{\mathbf{V}}_{dn}} \tag{3.27}$$

To derive gradients with respect to $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$, we use the chain rule and make use of the following identity: given $\widetilde{m} = \mathbf{w}^T\mathbf{m} + w_0$ and $\widetilde{v} = \mathbf{w}^T\mathbf{V}\mathbf{w}$ with a

Algorithm 1 Gradients for ELBO

Gradients with respect $\boldsymbol{\gamma}_n$ (3.29)

$$\frac{\partial \underline{\mathcal{L}}_n}{\partial \mathbf{V}_n} = \tfrac{1}{2}\left(\mathbf{V}_n^{-1} - \boldsymbol{\Sigma}^{-1}\right) + \sum_{d=1}^{D} \mathbf{W}_d^T \mathbf{G}_{dn}^v \mathbf{W}_d \tag{3.30}$$

$$\frac{\partial \underline{\mathcal{L}}_n}{\partial \mathbf{m}_n} = -\boldsymbol{\Sigma}^{-1}(\mathbf{m}_n - \boldsymbol{\mu}) + \sum_{d=1}^{D} \mathbf{W}_d^T \mathbf{g}_{dn}^m \tag{3.31}$$

Gradients with respect $\boldsymbol{\theta}$ (3.32)

$$\frac{\partial \underline{\mathcal{L}}}{\partial \mathbf{W}_d} = \sum_{n=1}^{N} \mathbf{g}_{dn}^m \mathbf{m}_n^T + 2\mathbf{G}_{dn}^v \mathbf{W}_d \mathbf{V}_n \tag{3.33}$$

$$\frac{\partial \underline{\mathcal{L}}}{\partial \mathbf{w}_{0d}} = \sum_{n=1}^{N} \mathbf{g}_{dn}^m \tag{3.34}$$

$$\frac{\partial \underline{\mathcal{L}}}{\partial \boldsymbol{\mu}} = -\sum_{n=1}^{N} \boldsymbol{\Sigma}^{-1}(\mathbf{m}_n - \boldsymbol{\mu}) \tag{3.35}$$

$$\frac{\partial \underline{\mathcal{L}}}{\partial \boldsymbol{\Sigma}^{-1}} = -\frac{1}{2}\sum_{n=1}^{N} \left[\mathbf{V}_n - \boldsymbol{\Sigma} + (\mathbf{m}_n - \boldsymbol{\mu})(\mathbf{m}_n - \boldsymbol{\mu})^T\right] \tag{3.36}$$

scalar w_0, vectors $\mathbf{m}$ and $\mathbf{w}$, and a positive definite matrix $\mathbf{V}$, we have the following derivatives,

$$\frac{\partial \widetilde{m}}{\partial \mathbf{m}} = \mathbf{w}, \qquad \frac{\partial \widetilde{v}}{\partial \mathbf{V}} = \mathbf{w}\mathbf{w}^T, \qquad \frac{\partial \widetilde{m}}{\partial \mathbf{w}} = \mathbf{m}, \qquad \frac{\partial \widetilde{v}}{\partial \mathbf{w}} = 2\mathbf{V}\mathbf{w} \tag{3.28}$$

We give the final expressions for gradients in Algorithm 1, and skip the derivation since it is straightforward. The updates of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can be obtained in closed form, by setting above gradients to 0. These updates are given below,

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{m}_n, \qquad \boldsymbol{\Sigma} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{V}_n + (\mathbf{m}_n - \boldsymbol{\mu})(\mathbf{m}_n - \boldsymbol{\mu})^T \tag{3.37}$$

Updates of other quantities can be done using a gradient based approach, but can also be available in closed form depending on the LVB. We now show an example where the all the updates are available in closed form.

3.5.2 An Example of Variational Learning using ELBO

We now give an example of the variational learning using ELBO. We choose the Gaussian LGM for which we derived ELBO in Section 3.2. In this case, the expectation term $\mathbb{E}_{q(\eta_{dn}|\widetilde{\gamma}_{dn})}[\log p(y_{dn}|\eta_{dn}, \boldsymbol{\theta})]$ has an analytical expression given below.

$$-\tfrac{1}{2}\log(2\pi\psi_d) - \tfrac{1}{2}[(y_{dn} - \widetilde{m}_{dn})^2 + \widetilde{v}_{dn}]/\psi_d \tag{3.38}$$

Taking the gradients with respect to $\widetilde{m}_{dn}$ and $\widetilde{v}_{dn}$, we get the following expressions for gradients g^m_{dn} and g^v_{dn},

$$g^m_{dn} = (y_{dn} - \widetilde{m}_{dn})/\psi_d \quad , \quad g^v_{dn} = -1/(2\psi_d) \tag{3.39}$$

We substitute these in the generalized gradient expression of Algorithm 1, and simplify. Updates of $\boldsymbol{\gamma}_n$ can be obtained similar to the procedure described in Appendix A.4, and are given below. These updates constitute the expectation or the E-step of the algorithm.

$$\mathbf{V} = (\boldsymbol{\Sigma}^{-1} + \mathbf{W}^T\boldsymbol{\Psi}^{-1}\mathbf{W})^{-1} \tag{3.40}$$

$$\mathbf{m}_n = \mathbf{V}\left[\mathbf{W}^T\boldsymbol{\Psi}^{-1}(\mathbf{y}_n - \mathbf{w}_0) + \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right] \tag{3.41}$$

Similarly, for parameters other than ψ_d, the updates can be derived following the procedure described in Appendix A.4. The updates for ψ_d can be obtained as shown in Khan [2011]. These updates constitute the maximization or the M-step of the algorithm, and are shown below.

$$\mathbf{W} = \left[\sum_{n=1}^{N}(\mathbf{y}_n - \mathbf{w}_0)\mathbf{m}_n^T\right]\left[\sum_{n=1}^{N}\mathbf{V} + \mathbf{m}_n\mathbf{m}_n^T\right]^{-1} \tag{3.42}$$

$$\mathbf{w}_0 = \frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_n - \mathbf{W}\mathbf{m}_n \tag{3.43}$$

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{m}_n, \qquad \boldsymbol{\Sigma} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{V}_n + (\mathbf{m}_n - \boldsymbol{\mu})(\mathbf{m}_n - \boldsymbol{\mu})^T \tag{3.44}$$

$$\psi_d = \frac{1}{N}\sum_{n=1}^{N} y_{dn}^2 - (y_{dn} - w_{0d})\mathbf{w}_d^T\mathbf{m}_n - w_{0d}^2 \tag{3.45}$$

The above updates coincide with the exact EM algorithm derived by Ghahramani and Hinton [1996]. The EM algorithm is exact since ELBO is tight after every E-step since true posterior is a Gaussian. See Beal [2003] for more details on the approximate and exact EM algorithm.

Algorithm 2 EM for parameter learning in LGMs with Gaussian likelihood

1. Initialize $\boldsymbol{\theta}$.
2. Iterate until convergence, between E and M step.
   (a) E-step:
      i. Compute posterior covariance using Eq. 3.40 (do only once).
      ii. For all $n = 1$ to N, compute posterior mean using Eq. 3.41.
   (b) Compute the lower bound of Eq. 3.21 and check for convergence.
   (c) M-step: Update $\boldsymbol{\theta}$ as shown in Eq. 3.42-3.45

An attractive feature of Gaussian LGMs is a property of their posterior covariance that most of the other LGMs lack. The posterior covariance in Eq. 3.40 does not depend on the data and can be computed beforehand. This leads to a huge computational saving since this computation involves a matrix inversion and need not be repeated for every data point.

The complete EM algorithm is summarized in Algorithm 2. Computation cost of the EM algorithm is linear in N and D. The E-step involves one matrix inversion, few multiplications of two matrices of sizes $L \times D$ and $D \times L$ respectively, and N multiplications of a $L \times D$ matrix with a $D \times 1$ vector, making the cost of E-step to be $O(L^3 + DL^2 + NDL)$. Note that matrix inversion has to be done only once per E-step, contributing L^3 cost, as opposed to NL^3. Similarly, M-step involves few summations over n of complexity $O(NL^2)$ and $O(NDL)$ plus a matrix inversion, making the total cost of M-step to be $O(NL^2 + NDL)$. Memory cost of the whole algorithm is $O(L^2 + L\min(D, N))$, where the first term arises from storing the sufficient statistics $\sum_n \mathbf{V} + \mathbf{m}_n\mathbf{m}_n^T$ and the second term arises from storing $\sum_n \mathbf{y}_n\mathbf{m}_n^T$. For the second term, we can either store the sum which takes $O(LD)$ memory or we can store all $\mathbf{m}_n$ which takes $O(LN)$ memory.

3.6 Fast Convergent Variational Inference

In this section, we derive a fast convergence variational inference algorithm for LGMs where there is one latent variable z_{dn} per y_{dn}, e.g. Gaussian processes and latent Gaussian graphical models (see Fig. 1.2(c) and 1.3). Without loss of generality, we can assume that $\mathbf{W} = \mathbf{I}_D$ and $\mathbf{w}_{0d} = 0$ respectively, since these quantities can always be "absorbed" in $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.

The simplest approach for inference is to use gradient based methods to optimize $\mathbf{m}_n$ and $\mathbf{V}_n$ directly. A problem with this approach is that the number of variational parameters is quadratic, i.e. $O(D^2)$ (note that if there is one latent variable per dimension, then $L = D$ and the size of $\mathbf{V}_n$ is $D \times D$). Opper and Archambeau [2009] speculate that this perhaps may be the reason behind limited use of variational approximations. Our proposed algorithm reduces the number of variational parameters from $O(D^2)$ to $O(D)$. Our derivation also serves as a demonstration of how concavity allows us to borrow ideas from concave optimization literature to design computationally efficient algorithms. This section is based on Khan et al. [2012b].

First, we rewrite the lower bound. For notational simplicity, we drop the subscript n from quantities such as $\mathbf{y}_n, \mathbf{m}_n, \mathbf{V}_n$ etc. For derivational simplicity, we assume that the arguments of $\underline{f}(y, \widetilde{\gamma}, \alpha)$ are all scalars; extension to vector case is straightforward. The ELBO of Eq. 3.21 can be re-written as follows,

$$\tfrac{1}{2}\left[\log|\mathbf{V}\boldsymbol{\Sigma}^{-1}| - \mathrm{Tr}(\mathbf{V}\boldsymbol{\Sigma}^{-1}) - (\mathbf{m}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{m}-\boldsymbol{\mu}) + L\right] + \sum_{d=1}^{D} \underline{f}(y_d, \gamma_d, \boldsymbol{\alpha}_d) \tag{3.46}$$

Note that $\widetilde{\gamma}_d = \gamma_d = \{m_d, V_{dd}\}$, since $\mathbf{W}_d$ is identity and $\mathbf{w}_{0d}$ is zero. This simplifies LVB terms which now only depend on the diagonal of $\mathbf{V}$. This simplification has a direct effect on the form that $\mathbf{m}$ and $\mathbf{V}$ take at the maximum. This will become clear from the fixed point equations for $\mathbf{m}$ and $\mathbf{V}$, shown below.

$$-\boldsymbol{\Sigma}^{-1}(\mathbf{m}-\boldsymbol{\mu}) + \mathbf{g}^m = 0 \tag{3.47}$$

$$\tfrac{1}{2}\left(\mathbf{V}^{-1} - \boldsymbol{\Sigma}^{-1}\right) + \mathrm{diag}(\mathbf{g}^v) = 0 \tag{3.48}$$

Here, $\mathbf{g}^m$ and $\mathbf{g}^v$ are gradients of $\underline{f}$ with respect to $\mathbf{m}$ and $\mathrm{diag}(\mathbf{V})$ respectively. We see that, at the solution, $\mathbf{V}$ is completely specified if $\mathbf{g}^v$ is known. This property can be exploited to reduce the number of variational parameters.

Opper and Archambeau [2009] (and Nickisch and Rasmussen [2008]) propose a reparameterization to reduce the number of parameters to $O(D)$. From the fixed point equation, we note that at the solution $\mathbf{m}$ and $\mathbf{V}$ will have the following form,

$$\mathbf{V} = (\boldsymbol{\Sigma}^{-1} + \mathrm{diag}(\boldsymbol{\lambda}))^{-1} \tag{3.49}$$

$$\mathbf{m} = \boldsymbol{\mu} + \boldsymbol{\Sigma}\mathbf{a} \tag{3.50}$$

where $\mathbf{a}$ and $\boldsymbol{\lambda}$ with $\lambda_d > 0, \forall d$. At the maximum (but not everywhere), $\mathbf{a}$ and $\boldsymbol{\lambda}$ will be equal to $\mathbf{g}^m$ and $\mathbf{g}^v$ respectively. Instead of solving the fixed-point equations to obtain $\mathbf{m}$ and $\mathbf{V}$, we can reparameterize the lower bound with respect to $\mathbf{a}$ and $\boldsymbol{\lambda}$. Substituting 3.49 and 3.50 in the ELBO of Eq. 3.46 and after simplification using matrix inversion and determinant lemmas, we get the following new ELBO,

$$\tfrac{1}{2}\left[-\log(|\mathbf{B}_\lambda||\mathrm{diag}(\boldsymbol{\lambda})|) + \mathrm{Tr}(\mathbf{B}_\lambda^{-1}\boldsymbol{\Sigma}) - \mathbf{a}^T\boldsymbol{\Sigma}\mathbf{a}\right] + \sum_{d=1}^{D} \underline{f}(y_d, \gamma_d, \alpha_d) \quad (3.51)$$

with $\mathbf{B}_\lambda = \mathrm{diag}(\boldsymbol{\lambda})^{-1} + \boldsymbol{\Sigma}$. For a detailed derivation, see Nickisch and Rasmussen [2008]. Since the mapping between $\{\mathbf{a}, \boldsymbol{\lambda}\}$ and $\{\mathbf{m}, \mathbf{V}\}$ is one-to-one, we can recover the latter given the former. The one-to-one relationship also implies that the new objective function has a unique maximum. The new lower bound involves vectors of size D, reducing the number of variational parameters to $O(D)$.

The problem with this reparameterization is that the new lower bound is no longer concave, even though it has a unique maximum. To see this, take the 1-D case. We collect all the terms involving V from Eq. 3.46, except the LVB term, to define the function shown in Eq. 3.52. We substitute the reparameterization $V = (\Sigma^{-1} + \lambda)^{-1}$ to get a new function in Eq. 3.53. The second derivative of this function is shown in Eq. 3.54. Clearly, this derivative is negative for $\lambda < 1/\Sigma$ and non-negative otherwise, making the function neither concave nor convex.

$$f(V) = \tfrac{1}{2}[\log(V\Sigma^{-1}) - V\Sigma^{-1}] \quad (3.52)$$

$$f(\lambda) = \tfrac{1}{2}[-\log(1 + \Sigma\lambda) - (1 + \Sigma\lambda)^{-1}] \quad (3.53)$$

$$f''(\lambda) = \tfrac{1}{2}\left(\frac{\Sigma}{1 + \Sigma\lambda}\right)^2 (\Sigma\lambda - 1) \quad (3.54)$$

Note that the function is still unimodal and the maximum of (3.51) is equal to the maximum of (3.46). With the reparameterization, we loose the concavity and therefore the algorithm may have slow convergence. Our experimental results, shown later, confirm the slow convergence.

3.6.1 A Coordinate Ascent Approach

We now derive an algorithm that reduces the number of variational parameters to $O(D)$ while maintaining concavity. Our algorithm uses simple scalar fixed point updates to obtain the diagonal elements of $\mathbf{V}$, instead of directly

optimization with respect to the full matrix. The complete algorithm is shown in Algorithm 3.

We define $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$, and denote the diagonal of $\mathbf{V}$ by $\mathbf{v}$.

To derive the algorithm, we first note that the fixed point equation of Eq. 3.48 has an attractive property. At the solution, the off-diagonal elements of $\mathbf{V}^{-1}$ are going to be same as the off-diagonal elements of $\mathbf{\Omega}$, i.e. if we denote $\mathbf{K} := \mathbf{V}^{-1}$, then $K_{ij} = \Omega_{ij}$. We have to only find the diagonal elements of $\mathbf{K}$ to get full $\mathbf{V}$. This is difficult, however, since gradient $\mathbf{g}^v$ depends on $\mathbf{v}$.

We take the approach of optimizing each diagonal element K_{ii} fixing all others (and fixing $\mathbf{m}$ as well). We partition $\mathbf{V}$ as shown in left side of Eq. 3.55, indexing the last row by 2 and rest of the rows by 1. We consider a similar partitioning of $\mathbf{K}$ and $\mathbf{\Omega}$. Our goal is to compute v_{22} and k_{22} given all other elements of $\mathbf{K}$, but first we establish a relationship between them. Matrices $\mathbf{K}$ and $\mathbf{V}$ are related through the blockwise inversion, as shown below.

$$\begin{bmatrix} \mathbf{V}_{11} & \mathbf{v}_{12} \\ \mathbf{v}_{12}^T & v_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{11}^{-1} + \frac{\mathbf{K}_{11}^{-1}\mathbf{k}_{12}\mathbf{k}_{12}^T\mathbf{K}_{11}^{-1}}{k_{22}-\mathbf{k}_{12}^T\mathbf{K}_{11}^{-1}\mathbf{k}_{12}} & -\frac{\mathbf{K}_{11}^{-1}\mathbf{k}_{12}}{k_{22}-\mathbf{k}_{12}^T\mathbf{K}_{11}^{-1}\mathbf{k}_{12}} \\ -\frac{\mathbf{k}_{12}^T\mathbf{K}_{11}^{-1}}{k_{22}-\mathbf{k}_{12}^T\mathbf{K}_{11}^{-1}\mathbf{k}_{12}} & \frac{1}{k_{22}-\mathbf{k}_{12}^T\mathbf{K}_{11}^{-1}\mathbf{k}_{12}} \end{bmatrix} \tag{3.55}$$

From the right bottom corner, we have Eq. 3.56 which we simplify to get Eq. 3.57.

$$v_{22} = 1/(k_{22} - \mathbf{k}_{12}^T\mathbf{K}_{11}^{-1}\mathbf{k}_{12}) \tag{3.56}$$

$$k_{22} = \mathbf{k}_{12}^T\mathbf{K}_{11}^{-1}\mathbf{k}_{12} + 1/v_{22} \tag{3.57}$$

Now, we know from the fixed point Eq. 3.48 that optimal v_{22} and k_{22} satisfy Eq. 3.58 at the solution, where g_{22}^v is the gradient of $\underline{f}$ with respect to v_{22}. Define, $t_{22} := \mathbf{k}_{12}^T\mathbf{K}_{11}^{-1}\mathbf{k}_{12}$ and substitute value of k_{22} from Eq. 3.57 in Eq. 3.58 to get Eq. 3.59. The solution that satisfies this fixed point can be found by maximizing the function defined in Eq. 3.59.

$$0 = k_{22} - \Omega_{22} + 2g_{22}^v \tag{3.58}$$

$$0 = t_{22} + 1/v_{22} - \Omega_{22} + 2g_{22}^v \tag{3.59}$$

$$f(v) = \log(v) + (t_{22} - \Omega_{22})v + 2\underline{f}(m_{22}, v) \tag{3.60}$$

The function $f(v)$ is a strictly concave function and we can optimize it by iterating the following,

$$\begin{aligned} k_{22} &= \Omega_{22} - 2g_{22}^v \\ v_{22} &= 1/(k_{22} - t_{22}) \end{aligned} \tag{3.61}$$

We will refer to this as a fixed-point iteration.

Since all elements of $\mathbf{K}$, but k_{22}, are fixed, t_{22} can be computed before hand and need not be evaluated at every fixed-point iteration. In fact, we do not need to compute t_{22} explicitly, since we can obtain its value using Eq. 3.57: $t_{22} = k_{22} - 1/v_{22}$. We do this before starting the fixed-point iteration. Complexity of the fixed point iterations depends on the gradient evaluations g^v_{22}, so its complexity is $O(1)$.

After convergence of iterations Eq. 3.61, we update $\mathbf{V}$ using Eq. 3.55. It turns out that this update can be written as one rank updates, complexity of which is $O(D^2)$. We now show how to update $\mathbf{V}$ efficiently. Let us denote the new values after the fixed point iterations by k^{new}_{22} and v^{new}_{22} respectively. Denote the old values by k^{old}_{22} and v^{old}_{22}. We use the right top corner of Eq. 3.55 to get first equality in Eq. 3.62. Using Eq. 3.57, we substitute $k_{22} - t_{22} = 1/v_{22}$ to get the second equality. Similarly, we use the top left corner of Eq. 3.55 to get the first equality in Eq. 3.63, and use Eq. 3.57 and 3.62 to get the second equality.

$$\mathbf{K}_{11}^{-1}\mathbf{k}_{12} = -(k^{old}_{22} - t_{22})\mathbf{v}^{old}_{12} = -\mathbf{v}^{old}_{12}/v^{old}_{22} \tag{3.62}$$

$$\mathbf{K}_{11}^{-1} = \mathbf{V}^{old}_{11} - \frac{\mathbf{K}_{11}^{-1}\mathbf{k}_{12}\mathbf{k}_{12}^T\mathbf{K}_{11}^{-1}}{k^{old}_{22} - t_{22}} = \mathbf{V}^{old}_{11} - \mathbf{v}^{old}_{12}(\mathbf{v}^{old}_{12})^T/v^{old}_{22} \tag{3.63}$$

Note that both $\mathbf{K}_{11}^{-1}$ and $\mathbf{k}_{12}$ do not change after the fixed point iteration. We use this fact to update $\mathbf{V}^{new}$. We use Eq. 3.55 to write updates for $\mathbf{V}^{new}$ and use 3.62 and 3.63 to simplify.

$$\mathbf{v}^{new}_{12} = \frac{\mathbf{K}_{11}^{-1}\mathbf{k}_{12}}{k^{new}_{22} - t_{22}} = -\frac{v^{new}_{22}}{v^{old}_{22}}\mathbf{v}^{old}_{12} \tag{3.64}$$

$$\mathbf{V}^{new}_{11} = \mathbf{K}_{11}^{-1} + \frac{\mathbf{K}_{11}^{-1}\mathbf{k}_{12}\mathbf{k}_{12}^T\mathbf{K}_{11}^{-1}}{k^{new}_{22} - t_{22}} \tag{3.65}$$

$$= \mathbf{V}^{old}_{11} - \frac{1}{v^{old}_{22}}\mathbf{v}^{old}_{12}(\mathbf{v}^{old}_{12})^T + \frac{v^{new}_{22}}{(v^{old}_{22})^2}\mathbf{v}^{old}_{12}(\mathbf{v}^{old}_{12})^T \tag{3.66}$$

$$= \mathbf{V}^{old}_{11} + \frac{v^{new}_{22} - v^{old}_{22}}{(v^{old}_{22})^2}\mathbf{v}^{old}_{12}(\mathbf{v}^{old}_{12})^T \tag{3.67}$$

After updating $\mathbf{V}$, we update $\mathbf{m}$ by optimizing the following non-linear least squares problem, cost of which is $O(D^2)$,

$$\max_{\mathbf{m}} -\tfrac{1}{2}(\mathbf{m} - \boldsymbol{\mu})^T\boldsymbol{\Omega}(\mathbf{m} - \boldsymbol{\mu}) + \sum_{n=1}^{D} \underline{f}(y_d, m_d, v_d, \alpha_d) \tag{3.68}$$

We use LBFGS algorithm for this.

Algorithm 3 Fast-convergent coordinate-ascent algorithm

1. Initialize $\mathbf{K} \leftarrow \mathbf{\Omega}, \mathbf{V} \leftarrow \mathbf{\Omega}^{-1}, \mathbf{m} \leftarrow \boldsymbol{\mu}$, where $\mathbf{\Omega} := \mathbf{\Sigma}^{-1}$.
2. Iterate until convergence, between updating columns of $\mathbf{V}$ and then $\mathbf{m}$ as shown below.
   (a) Update columns of $\mathbf{V}$ using the following.
      i. Rearrange $\mathbf{V}, \mathbf{K}, \mathbf{\Omega}$ so that the corresponding column is the last one.
      ii. $t_{22} \leftarrow k_{22} - 1/v_{22}$.
      iii. Store old value $v_{22}^{old} \leftarrow v_{22}$.
      iv. Iterate between $k_{22} \leftarrow \Omega_{22} - 2g_{22}^{v}$ and $v_{22} \leftarrow 1/(k_{22} - t_{22})$.
      v. Update $\mathbf{V}$.
         A. $\mathbf{V}_{11} \leftarrow \mathbf{V}_{11} + (v_{22} - v_{22}^{old})\mathbf{v}_{12}\mathbf{v}_{12}^{T}/(v_{22}^{old})^2$.
         B. $\mathbf{v}_{12} \leftarrow -v_{22}\mathbf{v}_{12}/v_{22}^{old}$.
   (b) Update $\mathbf{m}$ by maximizing the least squares problem of Eq. 3.68.

Computational complexity

The final algorithm is shown in Algorithm 3. The main advantage of our algorithm is its fast convergence as we show in the results section. Our algorithm is of similar flavor to graphical lasso [Friedman et al., 2008], and is in fact very close to a modified version presented in Mazumder and Hastie [2011]. Hence, our algorithm enjoys similar computational efficiency. Overall computational complexity is $O(D^3 + D^2 I^m + \sum_d I_d^f)$. First term is due to $O(D^2)$ update for each d. Second term is the cost of updating $\mathbf{m}$ where I^m is number of iterations required. Final term is for I_n^f iterations of fixed point updates, total cost linear in D due to summation. In all our experiments, I_d^f is usually 3 to 5, adding very little cost.

Proof of convergence

Proposition 2.7.1 in Bertsekas [1999] states that coordinate ascent algorithm will converge if the maximization with respect to each coordinate is uniquely attained. This is indeed the case for us since each fixed point iteration solves a concave problem of the form given in Eq. 3.60. Similarly, optimization with respect to $\mathbf{m}$ is also strictly concave. Hence, our algorithm converges to the maximum of the lower bound of Eq. 3.46.

Proof that V will always be positive definite

Let us assume that we start with a positive definite $\mathbf{K}$, which is true since we can initialize it with $\mathbf{\Omega}$. Now consider the update of v_{22} and k_{22}. Note that v_{22} is always going to be positive since it is the maximum of Eq. 3.60 which involves the log term. As $v_{22} > 0$, we get $k_{22} > \mathbf{k}_{12}^T \mathbf{K}_{11}^{-1} \mathbf{k}_{12}$ from Eq. 3.57. Hence, the Schur complement $k_{22} - \mathbf{k}_{12}^T \mathbf{K}_{11}^{-1} \mathbf{k}_{12} > 0$. Using this and the fact that $\mathbf{K}_{11}$ is positive definite, it follows that the new $\mathbf{K}$, after update of k_{22}, will be positive definite too. Hence, $\mathbf{V}$ will be positive definite too.

3.6.2 Results

We now show that the proposed algorithm lead to significant gain in speed on real problems. We consider the Gaussian process classification model for binary data using the Bernoulli logit link; see Section 1.2.3 for details on Gaussian process model. For Bernoulli logit link, we use the piecewise bound with 20 pieces, described in Section 4.5.

We apply the model on the UCI ionosphere data (available from UCI repository) which has 351 data examples with 34 features. We split the dataset keeping 80% of the dataset for training and rest for testing.

We compare our algorithm with the parameterization of Opper and Archambeau [Opper and Archambeau, 2009] (Eq. 3.51). We also compared to the naive method of optimizing with respect to full $\mathbf{m}$ and $\mathbf{V}$, e.g. method of Braun and McAuliffe [2010], but do not present these results since these algorithms have very slow convergence.

We examine the computational cost of each method in terms of the number of floating point operations (flops) for four hyperparameter settings $\boldsymbol{\theta} = \{\log(\sigma), \log(s)\}$. This comparison is shown in Fig. 3.1. The y-axis shows (negative of) the value of the lower bound, and the x-axis shows the number of flops. We draw markers at iteration 1,2,4,50 and in steps of 50 from then on. In all cases, due to non-convexity, the optimization of Opper and Archambeau reparameterization (black curve with squares) converges slowly, spending lot of time in flat regions of the objective functions and requiring a large number of computations to converge. The proposed algorithm (blue curve with circles) has consistently faster convergence than the other method. For this dataset, our algorithm always converged in 5 iterations.

We applied our method to two more datasets of Nickisch and Rasmussen [2008], namely 'sonar' (208 data example with 60 features) and 'usps' dataset (1540 data examples with 256 features), to observe similar behavior. Both of

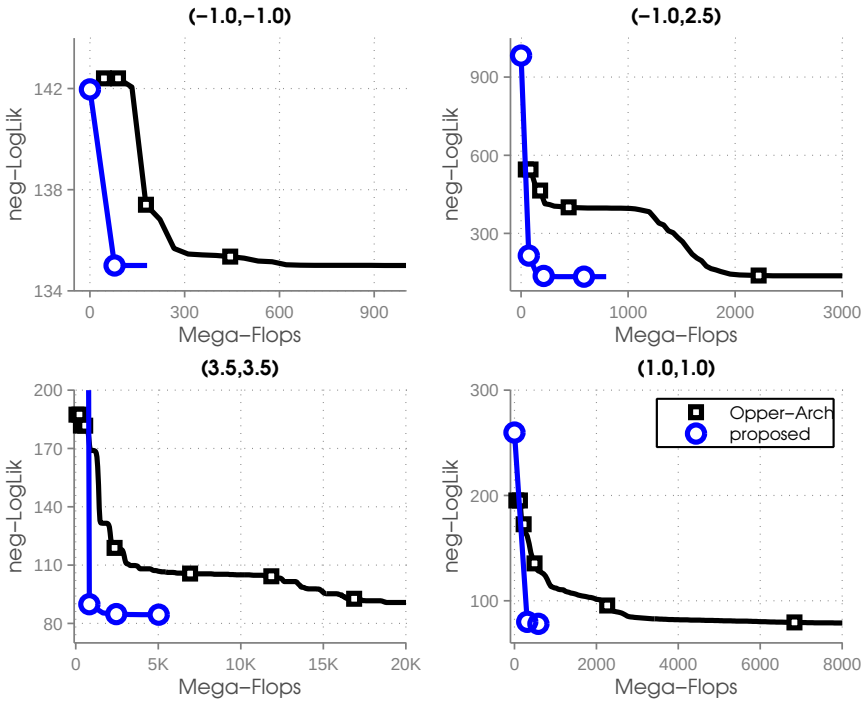


Figure 3.1: Convergence results on the binary ionosphere data set for Gaussian process classification. We plot the negative of the ELBO with respect to the number of flops. Each plot shows the progress of each algorithm for a hyperparameter setting shown at the top of the plot. The proposed algorithm always converges faster than the other method, in fact, in less than 5 iterations for this dataset.

these datasets are available at the UCI repository and details can be found in Nickisch and Rasmussen [2008].

We also compare the total cost to convergence in Table 3.1, where we count the total number of flops until successive increase in the objective function is below 10^{-3}. Each entry is a different setting of $\{\log(s), \log(\sigma)\}$. Rows correspond to values of $\log(s)$ while columns correspond for $\log(\sigma)$. Also, M,G,T stands for Mega, Giga, and Terra flops. We can see that the proposed algorithms takes much smaller number of operations compared to the existing algorithm.

Proposed Algorithm

	-1	1	3
-1	6M	7M	7M
1	26M	20M	22M
3	47M	81M	75M

Opper and Archambeau

	-1	1	3
-1	20G	212G	6T
1	101G	24T	24T
3	38G	1T	24T

Table 3.1: This table shows the total number of floating point operations for both algorithms to converge to a tolerance of 1e-3. Rows correspond to values of $\log(s)$ while columns correspond for $\log(\sigma)$. Here, M,G,T stands for Mega, Giga, and Tera flops. We can see that the proposed algorithms takes much smaller number of operations compared to the existing algorithm.

Chapter 4

Variational Learning of Binary LGMs

In this chapter, we discuss tractable variational learning for binary data. We propose two new LVBs and compare them to an existing LVB called the Jaakkola bound. First bound is a simple quadratic bound, called the Bohning bound. This bound leads to a faster learning, but less accurate, algorithm than the Jaakkola bound. The second bound is a class of highly accurate piecewise linear/quadratic bounds. Errors in these bounds can be made arbitrarily small by increasing the number of pieces. Our results show that the error in LVBs has direct effect on the accuracy of variational learning. We prove theoretical results showing that the piecewise bounds are more accurate than the Jaakkola bound, while the Bohning bound is less accurate than the Jaakkola bound. In terms of computational complexity, however, the Bohning bound is the fastest while the Jaakkola bound and the piecewise bounds have similar computational complexity. We demonstrate these results on many datasets and models.

4.1 Bernoulli Logit LGMs

We consider modeling of the binary data using Bernoulli logit likelihood, defined below for $y \in \{0, 1\}$ with a scalar predictor η,

$$p(y|\eta) = e^{y\eta}/(1 + e^{\eta}) = \exp[y\eta - \text{llp}(\eta)] \tag{4.1}$$

where $\text{llp}(\eta) = \log(1 + \exp(\eta))$ is the logistic-log-partition (LLP) function. Using this likelihood, we define a Bernoulli logit LGM (bLGM) to model binary data vectors $\mathbf{y}_n$,

$$p(\mathbf{z}_n|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{4.2}$$

$$\eta_{dn} = \mathbf{w}_d^T \mathbf{z}_n + w_{0d} \tag{4.3}$$

$$p(\mathbf{y}_n|\mathbf{z}_n, \boldsymbol{\theta}) = \prod_{d=1}^{D} \exp(y_{dn}\eta_{dn} - \text{llp}(\eta_{dn})) \tag{4.4}$$

The parameter set $\boldsymbol{\theta}$ is the set of parameters required to define the following quantities, $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{W}, \mathbf{w}_0\}$, where $\mathbf{W}$ is a matrix containing $\mathbf{w}_d^T$ as rows and $\mathbf{w}_0$ is a vector of w_{0d}. All other quantities are defined similar to the generic LGM of Section 1.1.

4.2 LVBs for bLGMs

As discussed before in Section 3.3, the ELBO is intractable since the expectation of the log-likelihood for discrete data is intractable. To be precise, we would like to bound the expectation of the log-likelihood of the observation y with respect to the approximate distribution $q(\eta|\widetilde{\gamma}) = \mathcal{N}(\eta|\widetilde{m}, \widetilde{v})$, i.e. the following: $\mathbb{E}_{q(\eta|\widetilde{\gamma})}[\log p(y|\eta)]$. This term is simplified below for the Bernoulli logit likelihood to show that the source of intractability is the expectation of the LLP function.

$$\mathbb{E}_{q(\eta|\widetilde{\gamma})}[\log p(y|\eta)] = y\widetilde{m} - \mathbb{E}_{q(\eta|\widetilde{\gamma})}[\mathrm{llp}(\eta)] \tag{4.5}$$

To obtain a tractable LVB, we need a tractable bound to the expectation of the LLP function. In this chapter, we discuss different ways to bound the LLP function. We describe the following LVBs in next few sections: the Jaakkola, Bohning, and piecewise bounds. All these bounds are jointly concave with respect to $\widetilde{m}$ and $\widetilde{v}$.

4.3 The Jaakkola Bound

The Jaakkola bound, proposed by Jaakkola and Jordan [1996], is a quadratic bound and is defined below,

$$\underline{f}^J(y, \widetilde{\gamma}, \xi) := y\widetilde{m} - \tfrac{1}{2}a_\xi(\widetilde{m}^2 + \widetilde{v}) - \tfrac{1}{2}\widetilde{m} - c_\xi \tag{4.6}$$

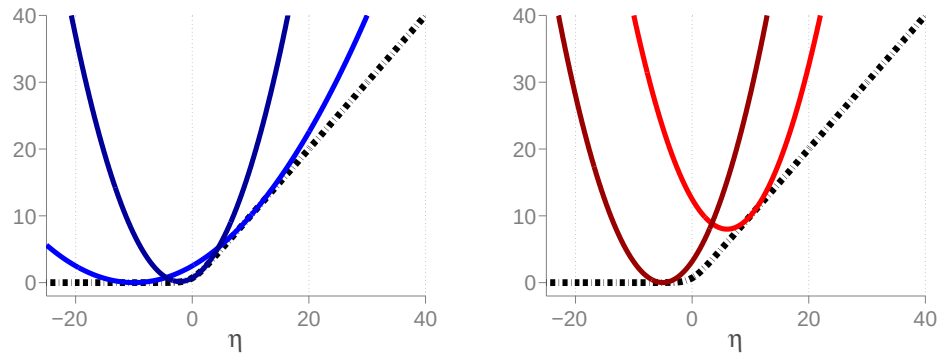
$$a_\xi = 2\lambda_\xi \tag{4.7}$$

$$c_\xi = -\lambda_\xi \xi^2 - \xi/2 + \mathrm{llp}(\xi) \tag{4.8}$$

$$\lambda_\xi = [g(\xi) - 1/2] / (2\xi) \tag{4.9}$$

where ξ is the variational parameter and a_ξ, c_ξ and λ_ξ are its functions. The Jaakkola bound is derived using an upper quadratic bound to the LLP function obtained by applying the Fenchel duality. Detailed derivation is given in the Appendix A.3.

Fig. 4.3 illustrates the quadratic bound to the LLP function for two values of ξ. The main difference between the Jaakkola bound and the Bohning bound (which we describe in the next section) is that the former allows



(a) The Jaakkola bound (b) The Bohning bound

Figure 4.1: This figure shows the upper quadratic bounds to the LLP function. Left plot shows the Jaakkola bound (in solid blue lines) for two values of ξ, along with the LLP function (in dashed black lines). The right plot shows the same for the Bohning bound (but in solid red lines) for two values of ψ. Note that the Bohning bound has fixed curvature, while the Jaakkola bound allows variable curvature thereby giving a more accurate bound. However, fixed curvature leads to a computationally cheaper algorithm.

variable curvature a_ξ. We will soon see that, for this reason, the Jaakkola bound is always more accurate than the Bohning bound, but this gain in accuracy comes with an increase in the computational cost.

4.3.1 Variational Learning

For a tractable lower bound, we need to bound $\mathbb{E}_{q(\eta|\widetilde{\gamma}_{dn})}[\log p(y_{dn}|\eta)]$ for the (d,n)'th observation. The Jaakkola bound to this term is shown below, where $a_{\xi,dn}$ and $c_{\xi,dn}$ are functions of the local variational parameter ξ_{dn} and are defined as in Eq. 4.7 and 4.8.

$$\underline{f}^J(y_{dn}, \widetilde{\gamma}_{dn}, \xi_{dn}) := y_{dn}\widetilde{m}_{dn} - \tfrac{1}{2}a_{\xi,dn}(\widetilde{m}_{dn}^2 + \widetilde{v}_{dn}) - \tfrac{1}{2}\widetilde{m}_{dn} - c_{\xi,dn} \tag{4.10}$$

The gradients with respect to $\widetilde{m}_{dn}$ and $\widetilde{v}_{dn}$ are given below,

$$g_{dn}^m = (y_{dn} - \tfrac{1}{2}) - a_{\xi,dn}\widetilde{m}_{dn} \quad , \quad g_{dn}^v = -a_{\xi,dn}/2 \tag{4.11}$$

We now derive updates for $\boldsymbol{\gamma}_n$, $\boldsymbol{\theta}$, and ξ_{dn}.

Updates for the posterior distribution and parameters

We substitute the expressions for g^m_{dn} and g^v_{dn} in the generalized gradient expressions given in Algorithm 1. We set the gradients to zero and simplify to get the updates which we describe below. Detailed derivation is given in Appendix A.4.

The E-step updates are shown below,

$$\mathbf{V}_n = \left(\boldsymbol{\Sigma}^{-1} + \mathbf{W}^T \mathbf{A}_n \mathbf{W}\right)^{-1} \tag{4.12}$$

$$\mathbf{m}_n = \mathbf{V}_n \left[\mathbf{W}^T(\mathbf{y}_n - \tfrac{1}{2} - \mathbf{A}_n \mathbf{w}_0) + \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right] \tag{4.13}$$

where $\mathbf{A}_n = \text{diag}(a_{\xi,1n}, a_{\xi,2n}, \ldots, a_{\xi,Dn})$. The M-step updates are the following,

$$\mathbf{w}_d^T = \left[\sum_{n=1}^{N}(y_{dn} - \tfrac{1}{2} - a_{\xi,dn}\mathbf{w}_0)\mathbf{m}_n^T\right]\left[\sum_{n=1}^{N} a_{\xi,dn}\left(\mathbf{V}_n + \mathbf{m}_n\mathbf{m}_n^T\right)\right]^{-1} \tag{4.14}$$

$$w_{0d} = \frac{\sum_{n=1}^{N} y_{dn} - \frac{1}{2} - a_{\xi,dn}\mathbf{w}_d^T\mathbf{m}_n}{\sum_{n=1}^{N} a_{\xi,dn}} \tag{4.15}$$

along with updates of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ which remains same as Eq. 3.37. For $\boldsymbol{\mu} = 0$ and $\boldsymbol{\Sigma} = \mathbf{I}$, these updates reduce to binary PCA discussed by Tipping [1998].

There are two complications to note in these updates. Firstly, the posterior covariance $\mathbf{V}_n$ depends on the data vector through $\mathbf{A}_n$ and needs to be computed for every n. Secondly, the updates for $\mathbf{w}_d$ and w_{0d} need to be done separately for each d, each of which involves reweighting the matrix $\sum_n \mathbf{V}_n + \mathbf{m}_n\mathbf{m}_n^T$ with $a_{\xi,dn}$.

These complications are direct consequences of the variable curvature of the Jaakkola bound. To see this, recall that the algorithm for the Gaussian likelihood, discussed in Section 3.5.2, had nice computational properties due to the fact that the noise variance ψ_d did not depend on n. The lower bound obtained using the Jaakkola bound corresponds to a Gaussian likelihood as shown in the following: $p(y_{dn}|\eta) \geq Z_{\xi,dn}\mathcal{N}(\tilde{y}_{dn}|\eta, 1/a_{\xi,dn})$ where $\tilde{y}_{dn} := (y_{dn} + 1/2)/a_{\xi,dn}$ and $Z_{\xi,dn}$ is a function of ξ_{dn}. This expression can be obtained using the quadratic upper bound given in Eq. A.25. The noise variance of the Gaussian lower bound depends on n, implying that the posterior distribution $\mathbf{V}_n$ also depends on n. Hence, we lose the nice computational properties we could have obtained by using a Gaussian lower bound with fixed noise variances. We will soon propose a new lower bound that simplifies the updates by using such lower bounds.

Update for the local variational parameter

We now discuss update of ξ_{dn}. For simplicity, we drop the subscript dn. We differentiate $\underline{f}^J$ with respect to ξ in Eq. 4.16, and simplify further in Eq. 4.17 by substituting the derivatives of a_ξ and c_ξ expressed in terms of the derivative of λ_ξ. We substitute the value of λ_ξ to get Eq. 4.18 and simplify to get the expression in Eq. 4.19.

$$\frac{\partial \underline{f}^J}{\partial \xi} = -\tfrac{1}{2}(\tilde{m}^2 + \tilde{v})\frac{\partial a_\xi}{\partial \xi} - \frac{\partial c_\xi}{\partial \xi} \tag{4.16}$$

$$= -(\tilde{m}^2 + \tilde{v} - \xi^2)\frac{\partial \lambda_\xi}{\partial \xi} + 2\xi\lambda_\xi + \tfrac{1}{2} - g(\xi) \tag{4.17}$$

$$= -(\tilde{m}^2 + \tilde{v} - \xi^2)\frac{\partial \lambda_\xi}{\partial \xi} + g(\xi) - \tfrac{1}{2} + \tfrac{1}{2} - g(\xi) \tag{4.18}$$

$$= -(\tilde{m}^2 + \tilde{v} - \xi^2)\frac{\partial \lambda_\xi}{\partial \xi} \tag{4.19}$$

Setting the gradient to zero gives us the update $\xi = \sqrt{\tilde{m}^2 + \tilde{v}}$.

A variational EM algorithm

A variational EM algorithm for parameter estimation is shown in Algorithm 4. The computational complexity is summarized in Table 4.1. Computing $\mathbf{V}_n$ requires inverting a $L \times L$ matrix and multiplication $\mathbf{W}^T\mathbf{A}_n\mathbf{W}$, making complexity of this step $O(L^3 + DL^2)$. Computing $\mathbf{m}_n$ requires a multiplication of two matrices of sizes $L \times D$ and $D \times L$, plus multiplications of a $L \times D$ matrix with a $D \times 1$ vector for few iterations. This makes the total cost of the E-step to be $O(N(L^3 + DL^2)I)$, where I is the number of iterations taken for convergence in the E-step. The cost of M-step and the memory cost are equal to $O(L^3 + NL^2 + NDL)$ and $O(L^2 + L\min(N, D))$.

4.4 The Bohning Bound

The Bohning bound, proposed by Bohning [1992], is a lesser known quadratic bound and is defined below.

$$\underline{f}^B(y, \widetilde{\gamma}, \psi) := y\widetilde{m} - \frac{1}{8}(\widetilde{m}^2 + \widetilde{v}) + b_\psi\widetilde{m} - c_\psi \tag{4.20}$$

$$b_\psi = \psi/4 - g_\psi \tag{4.21}$$

$$c_\psi = \psi^2/8 - g_\psi\psi + \mathrm{llp}(\psi) \tag{4.22}$$

$$g_\psi = 1/(1 + \exp(-\psi)) \tag{4.23}$$

Algorithm 4 Variational EM using the Jaakkola Bound

1. Initialize $\boldsymbol{\theta}$.
2. Iterate until convergence, between E and M step.
   (a) E-step (or Inference step):
      i. For each n, initialize $\mathbf{m}_n \leftarrow \boldsymbol{\mu}$ and $\mathbf{V}_n \leftarrow \boldsymbol{\Sigma}$ and iterate,
         A. Update $\xi_{dn} \leftarrow \sqrt{\tilde{m}_{dn}^2 + \tilde{v}_{dn}}$.
         B. Compute $a_{\xi,dn}$ and $c_{\xi,dn}$, $\forall d$ using Eq. 4.7 and 4.8.
         C. Compute $\mathbf{V}_n$ and $\mathbf{m}_n$ using Eq. 4.12 and 4.13.
   (b) Compute the lower bound of Eq. 3.21 and check for convergence.
   (c) M-step: Update $\boldsymbol{\theta}$ using Eq. 4.14,4.15, and 3.37.

Here, $\psi \in \mathbb{R}$ is the local variational parameter and g_ψ, b_ψ and c_ψ are functions of ψ. The Bohning bound is derived by forming a quadratic bound to the LLP function using a Taylor series expansion around ψ. Derivation is given in the next section.

Fig. 4.3 illustrates the quadratic bound to the LLP function for two values of ψ. An important feature of the Bohning bound is its fixed curvature which allows us to obtain a fast algorithm. This is in contrast to the Jaakkola bound which allows variable curvature. Choice of the local variational parameter ψ depends on the distribution of the expectation. As we will show later, the optimal ψ is equal to $\widetilde{m}$ as expected since we would like the bound to be tight around the high density area.

4.4.1 Derivation

The Bohning bound is derived using a Taylor series expansion of the LLP function around $\psi \in \mathbb{R}$ shown in Eq. 4.24. Functions $g_\psi = 1/(1+\exp(-\psi))$ and $h_\psi = g_\psi(1-g_\psi)$ are the first and second derivatives of the LLP function, and the equality holds for some $\chi \in \mathbb{R}$ due to Taylor's theorem [Rudin, 2006]. An upper bound to the LLP function is found by replacing the second derivative term by an upper bound. The second derivative of the LLP function evaluated at χ is upper bounded by $1/4$, since g_ψ lies between 0 and 1 and therefore the product $g_\psi(1-g_\psi)$ lies between 0 and $1/4$. Using

this fact, we get the upper bound shown in Eq. 4.25.

$$\text{llp}(\eta) = \text{llp}(\psi) + g_\psi(\eta - \psi) + \frac{1}{2}h_\chi(\eta - \psi)^2 \tag{4.24}$$

$$\leq \text{llp}(\psi) + g_\psi(\eta - \psi) + \frac{1}{8}(\eta - \psi)^2 \tag{4.25}$$

We substitute this bound in Eq. 4.5 and rearrange to obtain the Bohning bound shown in Eq. 4.20-4.23.

4.4.2 Variational Learning

Similar to the Jaakkola bound, we need to bound $\mathbb{E}_{q(\eta|\widetilde{\gamma}_{dn})}[\log p(y_{dn}|\eta)]$ for the (d, n)'th observation. The Bohning bound to this term is shown below, where $b_{\psi,dn}$ and $c_{\psi,dn}$ are functions of the local variational parameter ψ_{dn} and are defined as in Eq. 4.21 and 4.22.

$$\underline{f}^B(y_{dn}, \widetilde{\gamma}_{dn}, \psi_{dn}) := y_{dn}\widetilde{m}_{dn} - \frac{1}{8}(\widetilde{m}_{dn}^2 + \widetilde{v}_{dn}) + b_{\psi,dn}\widetilde{m}_{dn} - c_{\psi,dn} \tag{4.26}$$

To compute gradients with respect to $\boldsymbol{\gamma}_n$ and $\boldsymbol{\theta}$, we need gradients of the bound with respect to $\widetilde{m}_{dn}$ and $\widetilde{v}_{dn}$, which are given below,

$$g_{dn}^m = (y_{dn} + b_{\psi,dn}) - \widetilde{m}_{dn}/4 \quad , \quad g_{dn}^v = -1/8 \tag{4.27}$$

We now derive updates for $\boldsymbol{\gamma}_n$, $\boldsymbol{\theta}$, and ψ_{dn}.

Updates for the posterior distribution and parameters

We follow the same procedure as the Jaakkola bound, to get the updates given below. We substitute the expressions for g_{dn}^m and g_{dn}^v in the generalized gradient expressions given in Algorithm 1. We set the gradients to zero and simplify to get the updates below. Details are given in Appendix A.4.

The E-step updates are shown below,

$$\mathbf{V} = \left(\boldsymbol{\Sigma}^{-1} + \mathbf{W}^T\mathbf{W}/4\right)^{-1} \tag{4.28}$$

$$\mathbf{m}_n = \mathbf{V}\left[\mathbf{W}^T(\mathbf{y}_n + \mathbf{b}_n - \mathbf{w}_0/4) + \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right] \tag{4.29}$$

where $\mathbf{b}_n$ is the vector of $b_{\psi,dn}, \forall d$. The M-step updates are the following,

$$\mathbf{W} = \left[\sum_{n=1}^{N}\{4(\mathbf{y}_n + \mathbf{b}_n) - \mathbf{w}_0\}\mathbf{m}_n^T\right]\left[\sum_{n=1}^{N}\mathbf{V} + \mathbf{m}_n\mathbf{m}_n^T\right]^{-1} \tag{4.30}$$

$$\mathbf{w}_0 = \frac{1}{N}\sum_{n=1}^{N}4(\mathbf{y}_n + \mathbf{b}_n) - \mathbf{W}\mathbf{m}_n \tag{4.31}$$

along with updates of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ which remains same as Eq. 3.37.

The Bohning bound leads to simple closed form updates. Most importantly, the update for posterior covariance $\mathbf{V}$ does not depend on the data and can be computed before hand. This is similar to the case of Gaussian LGM, discussed in Section 3.5.2, where this property leads to huge computational saving. This is not the only property that the Bohning bound shares with Gaussian LGMs. In fact, when we compare these updates to Eq. 3.40-3.45, we find that all of the updates for the Bohning bound are exactly same as those of Gaussian LGMs with data vectors $4(\mathbf{y} + \mathbf{b}_n)$ and noise variance $\psi_d = 4$. This is due to the fact that the Bohning bound lower bounds the log-likelihood with a quadratic function and thereby lower bounds the likelihood with a Gaussian likelihood. The lower bound for the (d, n)'th measurement is an unnormalized Gaussian such that $p(y_{dn}|\eta) \geq Z_{\psi,dn}\mathcal{N}(\tilde{y}_{dn}|\eta, 4)$ for some $Z_{\psi,dn}$ and $\tilde{y}_{dn} = 4(y_{dn} + b_{\psi,dn})$. Hence, by optimizing ψ_{dn}, we find the best Gaussian likelihood (with a fixed noise variance) that gives the tightest lower bound to the marginal likelihood. As discussed before, the Jaakkola bound also bounds the likelihood with an unnormalized Gaussian likelihood, but allows variable noise variances.

Update for the local variational parameter

We now discuss update of ψ_{dn}. For simplicity, we drop the subscript dn from all quantities. Only $\underline{f}^B$ depends on ψ and we differentiate this term to obtain updates of ψ. The derivative is simplified below.

$$\frac{\partial \underline{f}^B}{\partial \psi} = \tilde{m}\frac{\partial b_\psi}{\partial \psi} - \frac{\partial c_\psi}{\partial \psi} \tag{4.32}$$

$$= \tilde{m}\,(1/4 - h_\psi) - (\psi/4 - g_\psi - h_\psi\psi + g_\psi) \tag{4.33}$$

$$= \tilde{m}\,(1/4 - h_\psi) - (\psi/4 - h_\psi\psi) \tag{4.34}$$

$$= h_\psi(\psi - \tilde{m}) - (\psi - \tilde{m})/4 \tag{4.35}$$

$$= (h_\psi - 1/4)(\psi - \tilde{m}) \tag{4.36}$$

The maximum occurs at $\psi = \tilde{m}$, which can be verified by computing the second derivative. This is as expected since the bound should be tight around the high density region of the distribution. The second solution $h_\psi = 1/4 \Rightarrow \psi = 0$, showing that this is not the maximum.

A variational EM algorithm

The variational EM algorithm for parameter estimation is shown in Algorithm 5. The computational complexity is summarized in Table 4.1, where

Algorithm 5 Variational EM using the Bohning Bound

1. Initialize $\boldsymbol{\theta}$.
2. Iterate until convergence, between E and M step.
   (a) E-step (or Inference step):
      i. Compute posterior covariance using Eq. 4.28 (do only once).
      ii. For each n, initialize $\mathbf{m}_n \leftarrow \boldsymbol{\mu}$ and iterate the following,
         A. Update $\boldsymbol{\psi}_n \leftarrow \mathbf{W}\mathbf{m}_n + \mathbf{w}_0$.
         B. Compute $b_{\psi,dn}$ and $c_{\psi,dn}$, $\forall d$ using Eq. 4.21 and 4.22.
         C. Compute posterior mean $\mathbf{m}_n$ using Eq. 4.29.
   (b) Compute the lower bound of Eq. 3.21 and check for convergence.
   (c) M-step: Update $\boldsymbol{\theta}$ using Eq. 4.30,4.31, and 3.37.

it is also compared to the complexity of the Jaakkola bound and the exact EM algorithm for Gaussian LGM. The Bohning bound has lower complexity than the Jaakkola bound. Since $\mathbf{V}$ is same for all n, we only need to be compute it once during the E-step, instead of computing it for all n separately. This simplification leads to a huge computational saving since computation of $\mathbf{V}$ involves inversion of a square matrix of size L, making complexity of this step $O(L^3 + DL^2)$, which is independent of n. Computing $\mathbf{m}_n$ requires a multiplication of two matrices of sizes $L \times D$ and $D \times L$, plus N multiplications of a $L \times D$ matrix with a $D \times 1$ vector for few iterations. This makes the total cost of the E-step to be $O(L^3 + DL^2 + NDLI)$, where I is the number of iterations taken for convergence in the E-step. The cost of M-step and the memory cost remains same as the Jaakkola bound.

4.5 Piecewise Linear/Quadratic Bounds

The quadratic bounds can be quite inaccurate at times. This is due to the fact that the integration is over the whole range of the approximation and any single-piece quadratic function will have unbounded error relative to the log-likelihood. For this reason, we propose the use of piecewise linear/quadratic bounds, which have a finite maximum error that can be driven to zero by increasing the number of pieces.

An R-piece quadratic bound consists of R intervals defined by $R + 1$ threshold points $t_0, ..., t_R$ such that $t_r < t_{r+1}$, and R quadratic functions

$a_r x^2 + b_r x + c_r$. An R-piece linear bound is a special case where $a_r = 0$ for all r. We fix the first and last threshold points to $-\infty$ and ∞, respectively. We use $\boldsymbol{\alpha}$ to denote the complete set of bound parameters including the threshold points and quadratic coefficients, denoting each quadratic piece by $\overline{f}(\boldsymbol{\alpha}, x)$. The piecewise quadratic bound is expressed as sum of piecewise upper bounds $\overline{f}_r$ to the LLP function, as shown below in Eq. 4.37. Upper bound $\overline{f}_r$ is the integration of r'th quadratic piece over the r'th interval, as shown in Eq. 4.38.

$$\underline{f}^{PW}(y, \widetilde{\gamma}, \boldsymbol{\alpha}) := y\widetilde{m} - \sum_{r=1}^{R} \overline{f}_r(\widetilde{m}, \widetilde{v}, \boldsymbol{\alpha}) \tag{4.37}$$

$$\overline{f}_r(\widetilde{m}, \widetilde{v}, \boldsymbol{\alpha}) := \int_{t_{r-1}}^{t_r} (a_r x^2 + b_r x + c_r)\mathcal{N}(x|\widetilde{m}, \widetilde{v})dx \tag{4.38}$$

The parameters $\boldsymbol{\alpha}$ are set such that each quadratic piece $(a_r x^2 + b_r x + c_r)$ is an upper bound to the LLP function, making $\overline{f}_r$ upper bounds and ultimately $\underline{f}^{PW}$ a lower bound. We make sure that each quadratic piece is as close as possible to the LLP function. This is done by solving a minimax optimization problem to minimize the maximum error between the LLP function and the quadratic piece. We describe this in detail in Section 4.5.1, but an important consequence of this optimization is that the maximum error made by piecewise bounds is always bounded. Not only this, but the error can be made arbitrary small by increasing the number of pieces. Another advantage is that $\boldsymbol{\alpha}$ can be precomputed and stored, and need not be optimized within the variational algorithm. This simplifies the algorithm and reduces the computation. The only disadvantage is that the function $\overline{f}_r$ does not have an analytical form, but since its value and its gradients can be evaluated at a given $\widetilde{m}$ and $\widetilde{v}$, we can use gradient based method to optimize the resulting evidence lower bound.

4.5.1 Derivation

In this section, we describe the computation of parameters of quadratic pieces, as well as the intervals they are defined in, while making sure that each piece is an upper bound to the LLP function. The minimax optimal R-piece quadratic upper bound problem for the LLP function is defined in Eq. 4.39. The objective function is simply the maximum gap between the piecewise quadratic bound and the LLP function. The first constraint is required to ensure that each quadratic function is an upper bound over the interval it is defined on. The second constraint ensures that the thresholds

are monotonically increasing. The final constraint ensures that the curvature of each quadratic function is non-negative.

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \max_{r\in\{1,..,R\}} \max_{x\in I_r} a_r x^2 + b_r x + c_r - \text{llp}(x) \\ &\quad a_r x^2 + b_r x + c_r - \text{llp}(x) \geq 0 \quad \forall\, r, x \in I_r := [t_{r-1}, t_r] \\ &\quad t_r - t_{r-1} > 0 \qquad\qquad \forall r \in \{1,..,R\} \\ &\quad a_r \geq 0 \qquad\qquad \forall r \in \{1,..,R\} \end{aligned} \tag{4.39}$$

We now reformulate the problem to remove all of the constraints. The second and third constraints can be dealt with using trivial reparameterizations. The first constraint can be replaced with an equality, which can then be solved for c_r yielding $c_r = -\left(\min_{x\in I_r} a_r x^2 + b_r x - \text{llp}(x)\right)$. This substitution is essentially finding the minimum gap between the quadratic and the LLP function on each interval and setting it to zero. This converts any quadratic with positive curvature into an upper bound on the LLP function over the corresponding interval. The final unconstrained problem is given below.

$$\min_{\boldsymbol{\alpha}} \max_{r\in\{1,..,R\}} [\max_{x\in I_r} a_r x^2 + b_r x - \text{llp}(x)] - [\min_{x\in I_r} a_r x^2 + b_r x - \text{llp}(x)] \tag{4.40}$$

We also note that the number of parameters that must be optimized can be reduced by a factor of two by exploiting the symmetry of the LLP function. The joint optimization of the breakpoints and quadratic coefficients can somewhat simplified by noting that the LLP function satisfies the relationship $\text{llp}(-x) = \text{llp}(x) - x$. If the function $ax^2 + bx + c - \text{llp}(x)$ yields an error of ϵ at t, then it is easy to show that the function $ax^2 + (1-b)x + c - \text{llp}(x)$ will also yields an error of ϵ at $-t$. For an even number of pieces R, we can exploit this observation by setting the breakpoints to $[-t_{R/2}, -t_{R/2-1}, ..., -t_1, 0, t_1, ..., t_{R/2-1}, t_{R/2}]$, optimizing coefficients for the intervals $[0, t_1], ..., [t_{R/2-1}, t_{R/2}]$, and then applying the above relation to convert the quadratic on each positive interval into a quadratic on the corresponding negative interval with the same maximum error. The procedure is almost identical in the case of an odd number of pieces, except that 0 is removed from the set of break points. So we need only optimize coefficients on the half of the intervals and then use the relationship given above to derive coefficients for the quadratics on the other (strictly negative) half of intervals. This reduces the number of variables that need to be optimized by half and makes the application of derivative-free methods practical with up to 20 pieces.

The main difficulty with the optimization problem of Eq. 4.40 comes from the fact that the inner maximization and minimization problems apparently have no closed-form solutions. However, global solutions for both the maximization and minimization problems can be easily found by numerical optimization as the function $ax^2 + bx - \text{llp}(x)$ has at most three critical points for any choice of a and b. However, this means that the outer minimization must be conducted using a derivative-free optimization algorithm since the objective function itself involves solving a non-linear optimization problem. We use the classical Nelder-Mead method [Nelder and Mead, 1965] for this purpose.

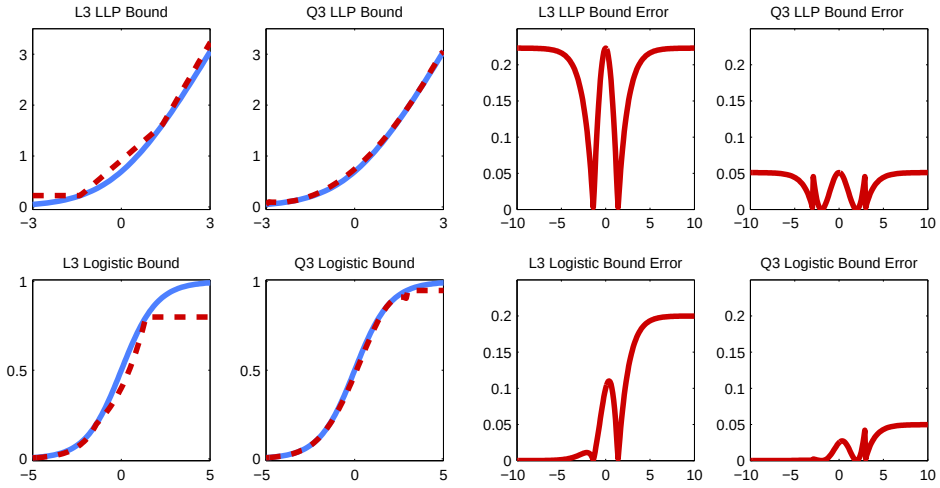
In the linear case where $a_r = 0$, [Hsiung et al., 2008] have proposed a constructive search method for determining minimax optimal coefficients and break points. Their work was motivated by the need to obtain linear approximations to LLP constraints in the context of geometric programming. We use their method for computing piecewise linear bounds. Whether a similar algorithm can be found for the piecewise quadratic case is an interesting open question that we leave for future work. The solutions found for the quadratic case using Nelder and Mead's method works well up to 20 pieces, which is more than sufficient for the applications we address in this thesis.

Fig. 4.2 illustrates the gain in accuracy obtained by using piecewise quadratic bounds instead of piecewise linear bounds. Fig. 4.2(a) and 4.2(b) contrast the accuracies obtained using three-piece linear and quadratic bounds while Fig. 4.3 shows the maximum error of both linear and quadratic bounds as a function of the number of pieces. We see that the piecewise quadratic bounds can be more than an order of magnitude more accurate than the piecewise linear bounds using the same number of pieces. Conversely, it can take more than double the number of pieces for a piecewise linear bound to approach the same accuracy as a piecewise quadratic bound.

4.5.2 Variational Learning

For variational learning, we need to evaluate $\underline{f}^{PW}$ and its gradients. These can be obtained using the moments of truncated Gaussians, since each $\overline{f}_r$ can be expressed in terms of those moments. See Appendix A.5 for details. Given the gradients of the piecewise bound, we can compute the gradients for variational learning using the generalized expressions given in Section 3.5.1. We can use any gradient based method, such as gradient-descent method, for optimization.

Computational cost of the algorithm is summarized in Table 4.1. These



(a) Bounds (b) Bounding errors

Figure 4.2: Figure (a) shows three-piece linear (L3) and quadratic (Q3) upper bounds. Top row shows these bounds on the LLP function and the bottom row shows the induced lower bounds on the Bernoulli logit likelihood. Figure (b) shows the corresponding error made in each plot in Figure (a).

computational costs are calculated according to the gradients shown in Algorithm 1. The gradients in E-step involves inversion of an $L \times L$ matrix, and multiplication of a $D \times L$ with $L \times D$ matrix, making the cost of each gradient step to be $O(L^3 + DL^2)$. For M-step, the most computational intensive step is the gradient of $\mathbf{W}_d$ which involves N multiplication of $D \times L$ matrix with $L \times D$ matrix, making the cost each gradient step in M-step to be $O(NDL^2)$. To compute the gradients in the M-step, we have to store all $\mathbf{m}_n$ and $\mathbf{V}_n$ making the memory cost to be $O(NL^2)$. This can be reduced to $L^2 + L\min(D, N)$ if we restrict M-step to take only one gradient step. The cost of gradients (g^m_{dn}, G^v_{dn}) for the piecewise bound scales linearly with the number of pieces R, adding DNR cost to both E and M steps.

4.6 Error Analysis

In this section, we compare the error obtained by local variational bounds. In terms of accuracy, the piecewise bound is the most accurate, followed by the Jaakkola bound and then the Bohning bound which is the least accurate of all. Our first theorem shows that the Jaakkola bound is always a better

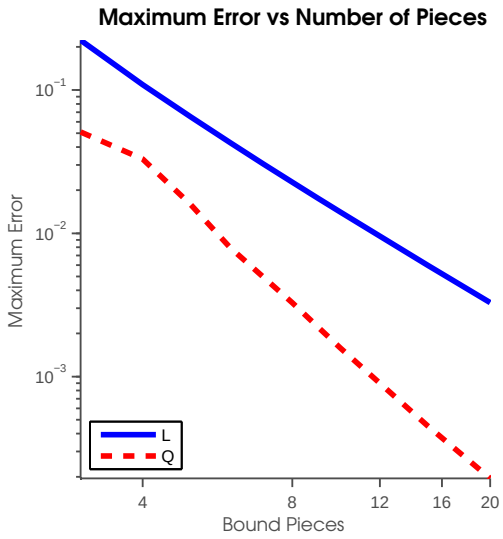


Figure 4.3: The maximum error in the LLP bounds as a function of the number of pieces in the bound. Here, 'L' stands for the linear bounds, while 'Q' stands for the quadratic bounds.

bound than the Bohning bound.

Theorem 4.6.1. *The Jaakkola bound is always more accurate than the Bohning bound for all* $\widetilde{m}$ *and* $\widetilde{v}$*.*

Proof. We derive an expression for the difference between the Jaakkola and Bohning bound, at their respective optimal variational parameter settings, and show that this difference is greater than 0, proving the superiority of the Jaakkola bound. The Bohning bound is optimized when $\psi^* = \widetilde{m}$. Substituting this in Eq. 4.20, we get the following optimal value of the Bohning bound,

$$f^B(y, \widetilde{\gamma}, \psi^*) = y\widetilde{m} - \frac{1}{8}(\widetilde{m}^2 + \widetilde{v}) + b_{\psi^*}\widetilde{m} - c_{\psi^*} = y\widetilde{m} - \widetilde{v}/8 - \text{llp}(\widetilde{m}) \quad (4.41)$$

Similarly, the Jaakkola bound is optimized when $\xi^* = \sqrt{\widetilde{m}^2 + \widetilde{v}}$, substituting which in Eq. 4.6 we get the optimal value of the Jaakkola bound,

$$f^J(y, \widetilde{\gamma}, \xi^*) = y\widetilde{m} - \tfrac{1}{2}a_{\xi^*}(\widetilde{m}^2 + \widetilde{v}) - \tfrac{1}{2}\widetilde{m} - c_{\xi^*} \quad (4.42)$$

$$= y\widetilde{m} - \widetilde{m}/2 + \sqrt{\widetilde{m}^2 + \widetilde{v}}/2 - \text{llp}\left(\sqrt{\widetilde{m}^2 + \widetilde{v}}\right) \quad (4.43)$$

Algorithm	E-step	M-step	Memory
Gaussian	$L^3 + DL^2 + NDL$	$L^3 + NL^2 + NDL$	L^2+ $L\min(D, N)$
Bohning	$L^3 + DL^2 + NDLI$	"	"
Jaakkola	$N(L^3 + DL^2)I$	"	"
Piecewise	$N(L^3 + DL^2 + DR)I$	$ND(L^2 + R)I$	NL^2

Table 4.1: Comparison of computational complexity. Each row is a variational EM algorithm. First row is the exact EM algorithm for Gaussian LGM described in Section 3.5.2. Next three rows are variational EM algorithm for bLGMs using various LVBs. The first two columns contain computational cost of E and M steps, while the third column contains the memory cost. All cost are in big O notation. I is the number of iterations required to converge. Note that the memory cost for piecewise bound can be reduced to $L^2 + L\min(D, N)$ by restricting M-step to one gradient step.

Taking the difference between the two bounds, we get the following,

$$\Delta(\widetilde{m}, \widetilde{v}) := f^J(y, \widetilde{\gamma}, \xi^*) - f^B(y, \widetilde{\gamma}, \psi^*) \tag{4.44}$$

$$= \widetilde{v}/8 + \text{llp}(\widetilde{m}) - \widetilde{m}/2 - \left[\text{llp}\left(\sqrt{\widetilde{m}^2 + \widetilde{v}}\right) - \sqrt{\widetilde{m}^2 + \widetilde{v}}\right] \tag{4.45}$$

Proving that this quantity is always greater than 0, will establish the superiority of the Jaakkola bound. To prove this, we first prove that the function $f(x) := \text{llp}(x) - x/2$ is monotonically decreasing. This can be established by noting that $x/2$ is a lower bound to $\text{llp}(x)$, and is asymptotically approaching $\text{llp}(x)$, making the difference smaller as $x \to \infty$. Using the monotonicity of $f(x)$ and noting that $\sqrt{\widetilde{m}^2 + \widetilde{v}} > \widetilde{m}$, we conclude that $f(\widetilde{m}) > f(\sqrt{\widetilde{m}^2 + \widetilde{v}})$. Since $\widetilde{v} > 0$, it follows that $\Delta(\widetilde{m}, \widetilde{v}) > 0, \forall \widetilde{m}, \widetilde{v}$, making the Jaakkola bound a better lower bound than the Bohning bound. □

As discussed before, the piecewise linear and quadratic bound have a known bounded maximum error ϵ_{max} by construction. This error can be made arbitrarily small by increasing the number of pieces. Hence, piecewise bounds can be made more accurate than any other bound, given enough number of pieces in the bound.

The fact that the piecewise bounds on the LLP function have a known finite maximum error ϵ_{max} means that we can easily bound the maximum error in the evidence lower bound. The following theorem formalizes this.

Theorem 4.6.2. *The loss in the evidence lower bound incurred by using the piecewise quadratic bound is at most* $D\epsilon_{\max}$. *In other words,* $\underline{\mathcal{L}}_n^J(\boldsymbol{\theta}, \boldsymbol{\gamma}_n) - \underline{\mathcal{L}}_n(\boldsymbol{\theta}, \boldsymbol{\gamma}_n, \boldsymbol{\alpha}) \leq D\epsilon_{\max}$ *for any* $\boldsymbol{\theta}, \boldsymbol{\gamma}_n$.

The proof is trivial. Since the error made in the d'th term is at most ϵ_{max}, the maximum error for all D dimension could not be more than $D\epsilon_{max}$.

We also note that the rate at which the error in the LLP bound decreases with number of pieces R is proportional to the rate at which $\underline{\mathcal{L}}_n$ approaches $\underline{\mathcal{L}}_n^J$. Hsiung et al. [2008] showed that the error in the optimal piecewise linear bound decreases with the approximate rate $\sqrt{2}/R^2$. The error in piecewise quadratic bounds decreases at least this fast. This means that $\underline{\mathcal{L}}_n^J - \underline{\mathcal{L}}_n$ approaches zero at a rate that is at least quadratic in the number of pieces.

4.7 Experiments and Results

In this section we compare different methods on several binary data sets. Throughout this section, we use $p(\mathbf{y}|\boldsymbol{\theta})$ to refer to the exact probability of a data vector $\mathbf{y}$ under the distribution with parameters $\boldsymbol{\theta}$. For small D, we can compute $p(\mathbf{y}|\boldsymbol{\theta})$ exactly using Monte Carlo, and compare the accuracy of various methods against it. In higher dimensions, we use imputation error as a measure of model fit. We hold out exactly one dimension per data case, selected at random. Given the observed data, we compute the prediction for the missing entries, and use the average cross-entropy of the held-out values as the imputation error measure.

Comparing errors in local variational bounds

In this experiment, we compare effects of the error in the local variational bounds. We do so by comparing the marginal likelihood estimates in a one-dimensional LGM with 1-D observation and 1-D latent variable.

We consider a binary latent Gaussian graphical model (LGGM) parameterized by a scalar mean μ and variance σ^2; see Fig. 1.3. The parameter vector is thus $\boldsymbol{\theta} = [\mu, \sigma^2]$. We set the true parameters $\boldsymbol{\theta}_*$ to $\mu_* = 2$ and $\sigma_* = 2$, yielding $p(y = 1|\boldsymbol{\theta}_*) = 0.7752$, which we denote by p_*.

Given $\boldsymbol{\theta}$, this probability can be estimated using various LVBs. Throughout the chapter, we discussed LVBs on the expectation of the log likelihood, but all of them imply an LVB on the log-likelihood itself. Consider the lower bound $\underline{f}(z)$ such that $\log p(y = 1|z) > \underline{f}(z)$. Using this, we can compute a

lower bound to $p(y=1|\boldsymbol{\theta})$ as follows,

$$p(y=1|\boldsymbol{\theta}) = \int p(y=1|z)\mathcal{N}(z|\mu,\sigma^2)dz \geq \int e^{\underline{f}(z)}\mathcal{N}(z|\mu,\sigma^2)dz \tag{4.46}$$

We denote this estimate as $\hat{p}(\boldsymbol{\theta})$. For all LVBs that we discuss, the above 1-D integral can be obtained in closed form. This is because all of the LVBs are (piecewise) linear/quadratic and application of the identity of Appendix A.1 gives a closed form expression.

Given $\boldsymbol{\theta}$, the log-marginal likelihood, in the limit of infinite data, can be computed as shown in Eq. 4.47. The log-marginal likelihood for θ_* can be computed as shown in Eq. 4.48. Finally, an estimate of the log-marginal likelihood can be obtained using the probability estimate $\hat{p}(\boldsymbol{\theta})$ as shown in Eq. 4.49.

$$\mathcal{L}(\boldsymbol{\theta}) = p_* \log p_\theta + (1-p_*)\log(1-p_\theta) \tag{4.47}$$

$$\mathcal{L}_* = p_* \log p_* + (1-p_*)\log(1-p_*) \tag{4.48}$$

$$\widehat{\mathcal{L}}(\boldsymbol{\theta}) = p_* \log \widehat{p}_\theta + (1-p_*)\log(1-\widehat{p}_\theta) \tag{4.49}$$

We compare these three quantities for various LVBs.

Note the two advantages of using these quantities for comparison. First, these quantities are computed in the limit of infinite data and hence there is no estimation error due to the data. Second, the marginal likelihood estimates do not require use of Jensen's inequality and therefore do not have any additional error due to it. Hence, the estimates are free from any other type of error and contain error introduced by LVBs only.

We fix μ to its optimal value and vary σ. The results of this experiment are given in Fig. 4.4. In each subplot, the solid lines show $\mathcal{L}(\boldsymbol{\theta})$ and the circle shows $\mathcal{L}_*$ for the true parameter value, i.e. $\sigma_* = 2$. We clearly see that the maximum occurs at the true parameter value, as desired. The dashed lines show the estimate $\widehat{\mathcal{L}}(\boldsymbol{\theta})$. We see that the Bohning (B) and Jaakkola (J) bounds fail dramatically, estimating $\sigma = 0$ instead of the correct value $\sigma = 2$. The piecewise bounds do significantly better, converging to the true marginal likelihood and correct σ value as the number of pieces in the bound increases. The piecewise quadratic bounds (Q3 and Q5) converge significantly faster than the linear bounds (L6 and L10), as predicted by the maximum error analysis in Section 4.6. Note that the results for Q3 and Q5 match those of L6 and L10, suggesting that the quadratic bound converges twice as fast as a function of the number of pieces.

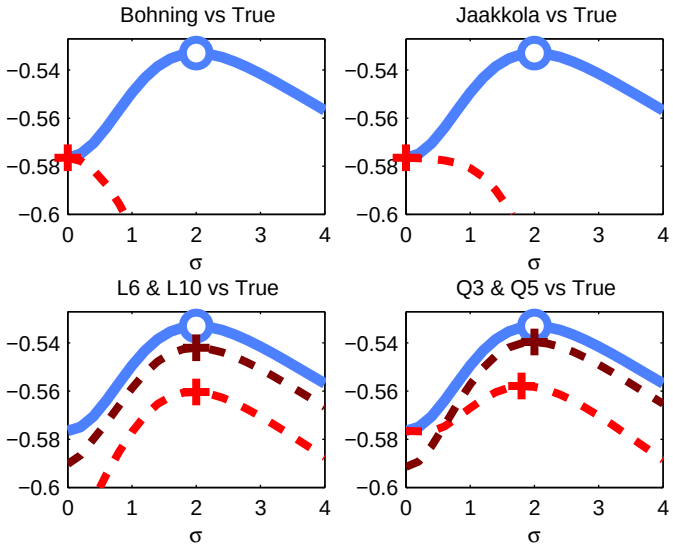
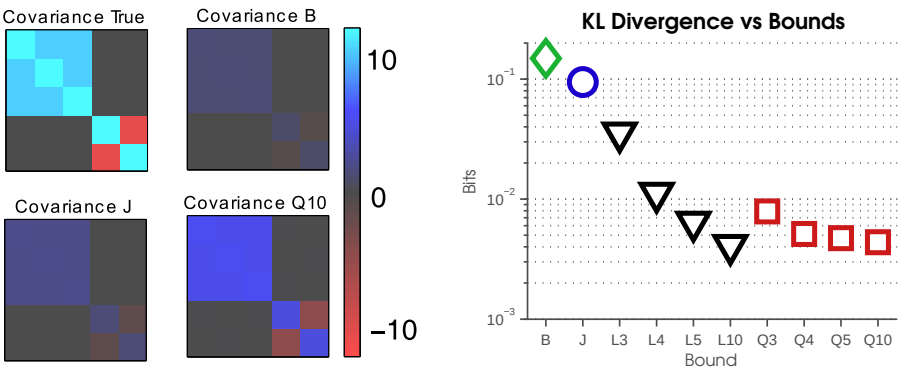


Figure 4.4: This figure shows results for the 1D synthetic LGGM experiment. We show the Bohning, Jaakkola, piecewise linear bounds with 6 and 10 pieces (denoted by L6 and L10 respectively), and piecewise quadratic bounds with 3 and 5 pieces (denoted by Q3 and Q5). The bounds are shown in red dashed lines with darker colors indicating more pieces. The true marginal likelihood is shown in blue solid lines. Markers show the true and estimated parameter values.

Comparing parameter estimates

In this experiment, we compare the accuracy of parameter estimates. We consider a 5D binary latent Gaussian graphical model (bLGGM) with known parameters. We set the true mean vector $\boldsymbol{\mu}_*$ to 0 and the true covariance matrix $\boldsymbol{\Sigma}_*$ as seen in the top left panel of Fig. 4.5(a). With these parameter setting, we get data vectors where first 3 dimensions have high positive correlation and last 2 dimensions have high negative correlations, the two blocks being independent of each other. Similar to the previous experiment, we compute the true distribution $p(\mathbf{y}|\boldsymbol{\theta}_*)$ for all 32 binary data vectors. We can compute these values to a reasonable accuracy using the samples from the true model since the latent-dimensionality is small (we use 10^6 samples). However, unlike the previous experiment, we estimate $\boldsymbol{\theta}$ by optimizing the evidence lower bound described in Chapter 3. Consequently, in this experiment, there is no error due to the data, but due to the application of the



(a) 5D bLGGM covariance

(b) 5D bLGGM KL divergence

Figure 4.5: Figure (a) shows the true covariance matrix for the synthetic bLGGM experiment along with the covariance estimates using the Bohning, Jaakkola, and 10 piece quadratic bounds, indicated with 'B','J', and 'Q10' respectively. Figure (b) shows the KL divergence between the true and estimated distributions for the 5D synthetic bLGGM experiment. We show results for the Bohning and Jaakkola bounds, as well as $3, 4, 5$ and 10 piece linear and quadratic bounds.

Jensen inequality and the LVBs.

Fig. 4.5(a) shows the covariance matrices estimated using the Jaakkola (J), Bohning (B) and 10 piece quadratic bounds (Q10). We see that both Bohning and Jaakkola shrink the estimated covariance parameters considerably, while the 10 piece quadratic bound results in less biased parameter estimates. Fig. 4.5(b) shows the KL divergence between $p(\mathbf{y}|\boldsymbol{\theta}_*)$ and $p(\mathbf{y}|\hat{\boldsymbol{\theta}})$ for the parameters $\hat{\boldsymbol{\theta}}$ estimated using each bound; both quantities estimated using the samples from the model, as discussed earlier. We show results for Bohning (B), Jaakkola (J) and 3 to 10 piece linear and quadratic bounds (L3-L10,Q3-Q10). We see that the piecewise bounds have significantly lower KL divergence than the Bohning and Jaakkola bounds when using a sufficient number of pieces. This indicates that they estimate significantly more accurate models, as suggested by the covariance plots in Fig. 4.5(a). We again see that the piecewise quadratic bound converges approximately twice as fast as the piecewise linear bound as a function of the number of pieces.

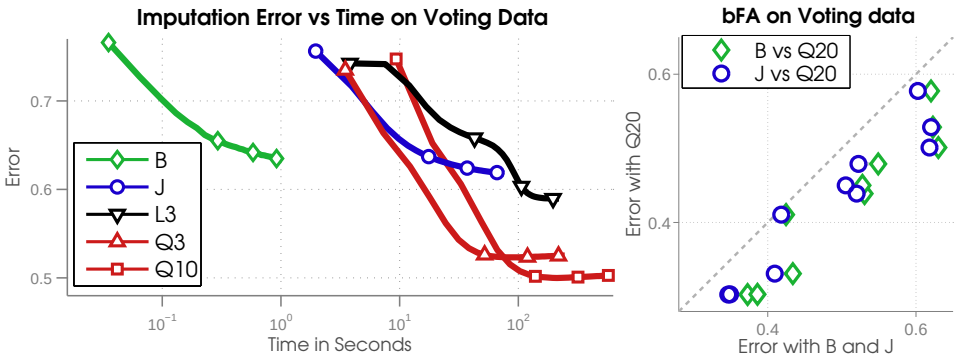


Figure 4.6: Results for bFA on the Voting data: Left plot shows the imputation error versus time on the UCI Voting data. Markers are plotted at iterations $2, 10, 20, 35$. We see that the piecewise bound gives much lower error and takes a times comparable to the Jaakkola bound. Right plot shows the imputation error of the 20-piece quadratic bound relative to Bohning and Jaakkola for the FA model. Each point is a different train-test split and a point below the dashed line indicates that piecewise bound performs better than other bounds.

Results for binary factor analysis (bFA)

We fit a three-factor binary factor analysis (bFA) model to the congressional voting records data set (available in the UCI repository) which contains votes of 435 U.S. Congressmen on 16 issues and the party of congressmen. We remove the data points which contain missing values and 3 issues which only show mild correlation with other issues. This gives us a total of 258 data vectors with 14 dimensions each (13 issues plus the party of the congressman). We use 80% of the data for training and 20% for testing.

Fig. 4.6 (left) shows traces of the imputation error versus time for Jaakkola (J), Bohning (B), three-piece linear (L3) and three and ten piece quadratic bounds (Q3, Q10) for one training-test split. We see that the piecewise bounds give lower error than the Jaakkola and Bohning bounds, but require more time to converge. We again observe that the quadratic bounds have lower error than the linear bounds and the error decreases as the number of pieces increases. Fig. 4.6 (right) shows the final imputation error results for 10 training-test splits. We plot the error of Q20 against that of B and J. We clearly see that Q20 outperforms both B and J for all splits.

Fig. 4.7-4.9 illustrate the results obtained using a 2-factor model on the

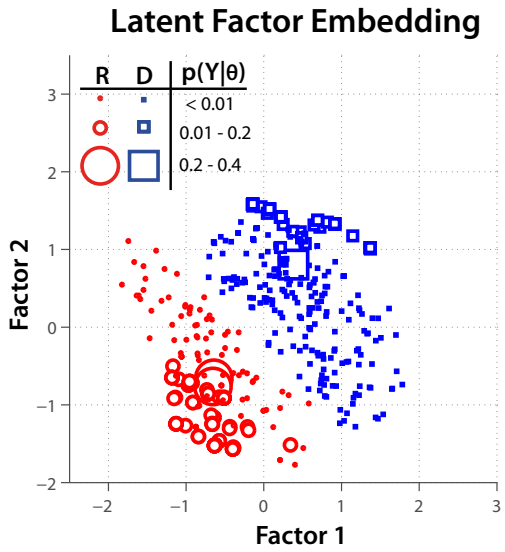


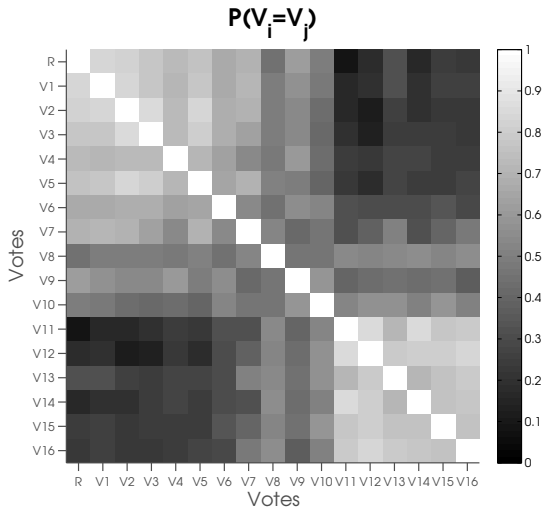
Figure 4.7: Results for 2-factor bFA on the voting data using the piecewise bound. This figure shows a plot of posterior means of factors. Each point represents a congressman, with size of the marker proportional to the value of the marginal likelihood; see legend for details. Republicans (R) are marked with circles while Democrats (D) are marked with squares.

voting data with the Q20 piecewise bound. Fig. 4.7 shows posterior means of factors. Each point represents a congressman, with size of the marker proportional to the value of the marginal likelihood approximation; see the legend for details on size. Republicans (R) are marked with circles while Democrats (D) are marked with squares. We see that the factors are nicely clustered, clearly bringing out the fact that the Republicans and Democrats have different voting patterns. Also note that, in each cluster, there are only few congressmen with large marginal likelihoods (the big markers). These congressmen, perhaps the most "consistent" Republicans/Democrats, represent the voting pattern of the whole party, and are most discriminative in deciding the party type.

Left figure in Fig. 4.8 shows the names of the issues, while the right figure shows the probability of two issues getting the same vote. To be

	Description
V1	budget-resolution
V2	aid-nicaraguan-contras
V3	mx-missile
V4	duty-free-exports
V5	anti-satellite-test-ban
V6	handicapped-infants
V7	export-admin-south-africa
V8	immigration
V9	synfuels-corp-cutback
V10	water-project-cost-sharing
V11	physician-fee-freeze
V12	El-salvador-aid
V13	religious-groups-in-schools
V14	education-spending
V15	superfund-right-to-sue
V16	crime

(a)



(b)

Figure 4.8: Left Figure shows the names of the issues. Right figure shows the probability of two issues getting the same vote, computed according to Eq. 4.50.

precise, each point (i, j) in the plot represents the following probability,

$$p(V_i = V_j|\hat{\boldsymbol{\theta}}) \approx \int p(V_i = 1|\mathbf{z}, \hat{\boldsymbol{\theta}})p(V_j = 1|\mathbf{z}, \hat{\boldsymbol{\theta}})\mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})d\mathbf{z} + \int p(V_i = 0|\mathbf{z}, \hat{\boldsymbol{\theta}})p(V_j = 0|\mathbf{z}, \hat{\boldsymbol{\theta}})\mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})d\mathbf{z} \tag{4.50}$$

where V_i is the i'th vote/issue (equal to the i'th element of the data vector $\mathbf{y}$), and $\hat{\boldsymbol{\theta}}$ is the parameter estimate. We approximate the integral with Monte Carlo which is efficient since $\mathbf{z}$ is 2D. This probability represents the correlation in the issues. A high value indicates that if a voter votes 'yes' (or 'no') to one issue, she is more likely to vote 'yes' (or 'no') to the other issue as well. We see that the voting patterns are clustered – the two groups V1-V7 and V11-V16 are positively correlated among themselves, while being negatively correlated across the two groups.

Finally, Fig. 4.9 shows the probability of voting 'yes' to an issue given the party of the congressman. We see a clear partisan behavior for the two groups V1-V7 and V11 to V16, where whenever Republicans vote 'yes' the Democrats vote 'no' and vice versa.

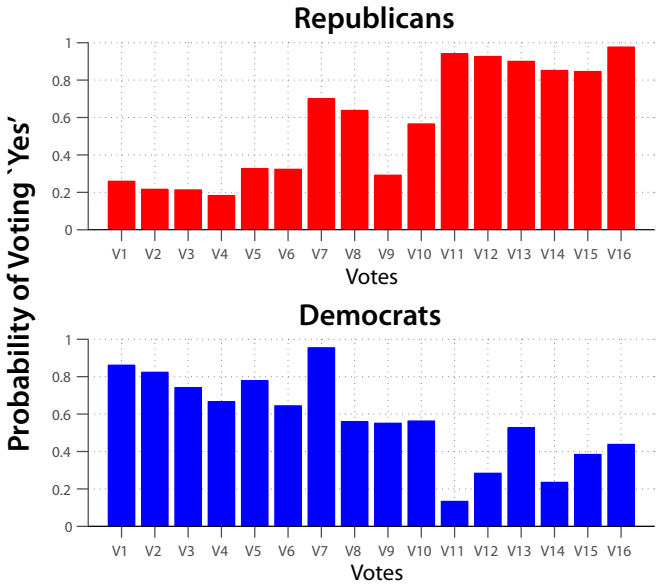


Figure 4.9: The probability of voting 'yes' to an issue given the party.

Results for binary latent Gaussian graphical model (bLGGM)

Next, we fit the binary LGGM (bLGGM) and sparse binary LGGM (sbLGGM) models to the UCI LED data set (available in the UCI repository). The sparse LGGM model is same as LGGM but now has a sparse $\mathbf{\Sigma}^{-1}$. This is achieved by assuming a sparse prior on this matrix (similar to the graphical lasso, see Friedman et al. [2008]). The EM algorithm remains almost the same, with a graphical lasso optimization in the M-step instead of a closed form update of $\mathbf{\Sigma}$. The derivations are straightforward and we do not give any details; interested readers should refer to the graphical lasso paper.

The LED data set has 2000 data cases and 24 dimensions. The data is synthetically generated but is out of the LGM model class. It contains 7 highly correlated variables that decide the LED display, and 17 other variables that take random values independent of the display. In the bLGGM experiment, we use 80% of the data for training and 20% for testing. The first plot in Fig. 4.10 shows the results for 10 training-test splits. As in the Voting data, Q20 outperforms both B and J on all splits.

In the sbLGGM experiment, we purposely under-sample the training set using 10% of the data for training and 50% for testing. The second plot in Fig. 4.10 shows the results for 10 train-test splits. We plot the error of Q20 versus B and J for the optimal choice of the regularization parameter

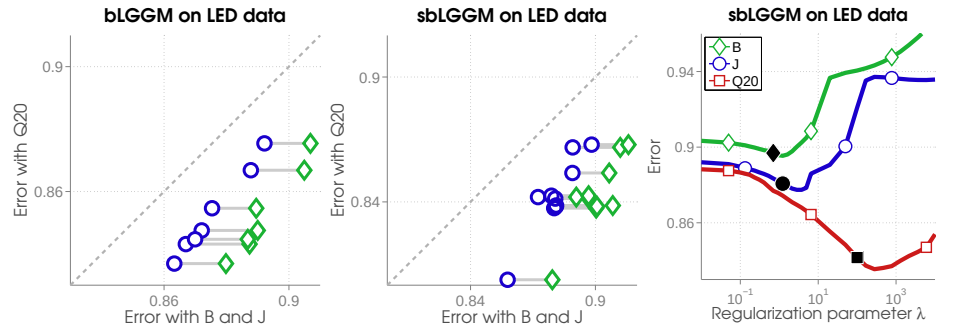


Figure 4.10: Results for bLGGM on the LED dataset. The first two plots, on the left, show the imputation error of the 20-piece quadratic bound relative to Bohning and Jaakkola for bLGGM and sbLGGM. Each point is a different train-test split and a point below the dashed line indicates that piecewise bound performs better than other bounds. The third plot on the right shows the imputation error versus the regularization parameter setting λ for sbLGGM.

λ (for sparse prior) found using cross-validation. We again see that Q20 outperforms both B and J on all splits. The final plot in Fig. 4.10 shows traces of the imputation error as a function of the regularization parameter setting for a single split. The optimal value of λ for each bound corresponds to precision matrices that are 82.6%, 83.7% and 80.4% sparse for B, J and Q20, respectively.

Fig. 4.11 shows the posterior mean at the optimal value of λ, while Fig. 4.12 shows the posterior covariance and precision matrices. As expected, all the methods show significant mean and correlations for the first 7 relevant variables, clearly showing that these variables are important for prediction. Note that B and J have significantly lower values for posterior mean and covariance than Q20. This behavior is similar to the results of the 5D synthetic data discussed earlier, and shows that B and J shrink the mean and covariance value significantly leading to poor performances.

Comparison with expectation propagation

In this section, we compare performance of expectation progression (EP) [Minka, 2001] with the variational approach using the piecewise bound. See Section 2.3 for details on EP. We consider fitting binary GP classification on the Ionosphere dataset. The dataset has been investigated previously in Kuss and Rasmussen [2005]; Nickisch and Rasmussen [2008] and we repeat

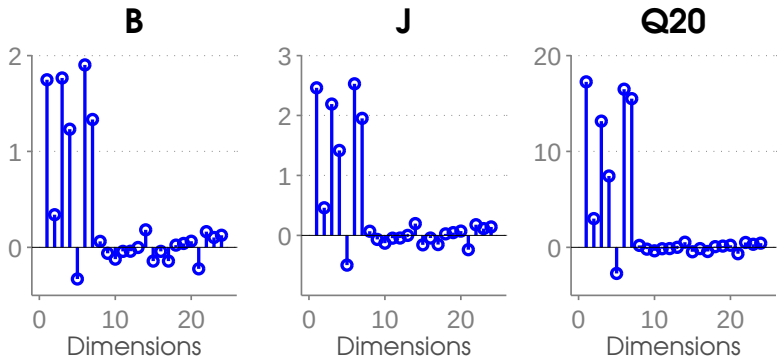


Figure 4.11: Posterior mean for the LED dataset at the optimal value of λ.

their experiments. This datasets consists of 351 instances of radar measurements from the ionosphere and the task is to classify these measurements as good or bad. We use 200 instances for training and rest for testing. We consider a squared-exponential kernel for the covariance parameter, under which (i, j)'th entry of $\mathbf{\Sigma}$ is defined as follows $\mathbf{\Sigma}_{ij} = -\sigma^2 \exp[-\frac{1}{2}||\mathbf{x}_i - \mathbf{x}_j||^2/s]$. We set $\boldsymbol{\mu} = 0$. The set of parameters is $\boldsymbol{\theta} = (\log(\sigma), \log(s))$.

We compare the approximate posterior distribution in Fig. 4.13. We consider two parameter settings. Fig. 4.13(a) shows posterior mean and covariance for $\log(s) = -1, \log(\sigma) = -1$. This parameter setting correspond to an easy inference problem, since the true posterior distribution for this setting is very close to a Gaussian distribution (for reasons explained in Kuss and Rasmussen [2005]). We plot elements of mean and covariance obtained using our approach vs the ones obtained by EP. We see that both approximations give almost identical posterior distributions for this easy case. Next, we consider a parameter setting for which posterior distribution is skewed, making it a difficult case. We set $\log(s) = -1, \log(\sigma) = 4$, results for which are shown in Fig. 4.13(b). We note that the two methods give different approximations for this parameter setting. EP matches the moments so it finds the mean and covariance which are very close to those of the true distributions. The estimates obtained with the variational method shows the expected behavior which results from minimizing the Jensen's lower bound [Bishop, 2006, Chapter 10]. The "zero-enforcing" property ensures that the mean is shifted away from zero and the variance is shrunk. The question is which approximation is better? We now show that for all purposes both approximations are equally good.

We compare several quantities of interest. First, we compute the Jensen's lower bound to the marginal likelihood obtained using both methods. Sec-

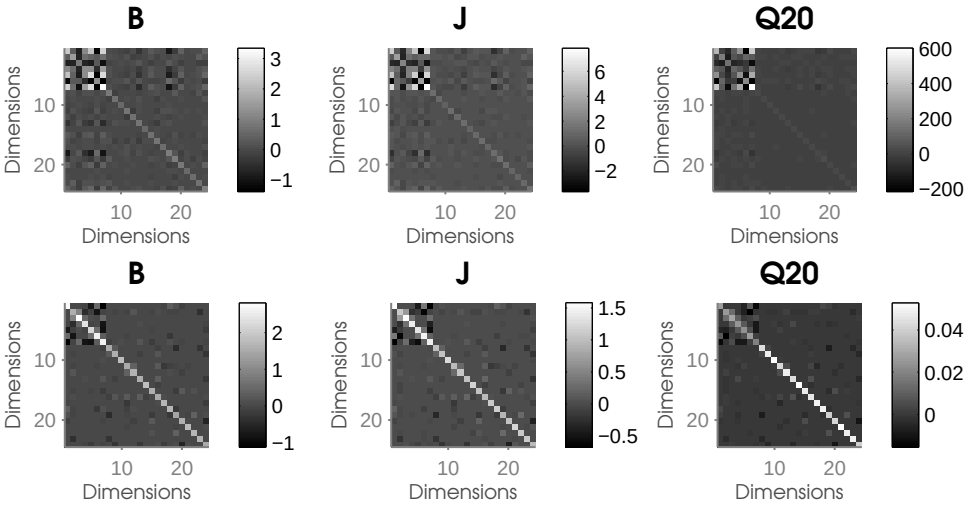
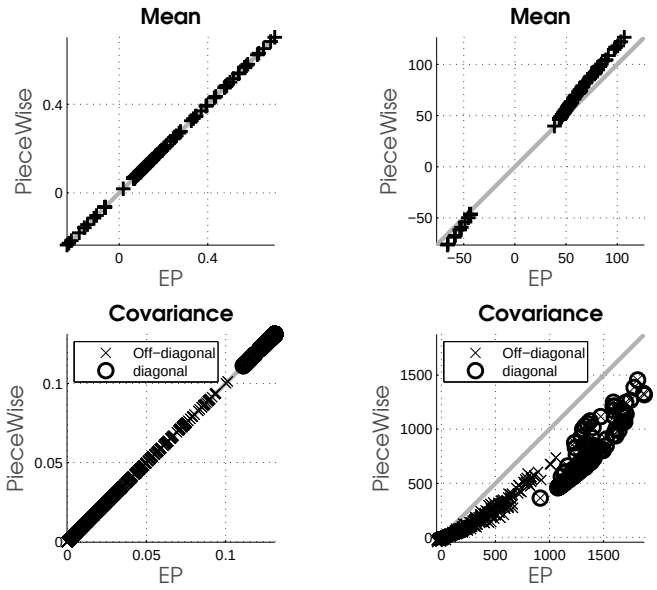


Figure 4.12: Top row shows the posterior covariance matrices for the LED dataset and bottom row shows the corresponding precision matrices, again at the optimal value of λ.

ond, we compute the EP approximation to the marginal likelihood for both the methods; see Eq. 2.32 in Section 2.3. Computation of the above quantity for our approach requires computation of site functions, for which we run one step of EP given the posterior mean and covariance obtained using our approach. Finally, we compute predictions and compare cross entropy prediction error for both methods. We plot these quantities for various values of $\log(\sigma)$ and $\log(s)$. Fig. 4.14 shows that both approaches give almost identical results for all quantities. Hence, the posterior distribution obtained by both methods are equally good for all the tasks.

We repeat this experiment on a larger USPS digit dataset. Similar to Kuss and Rasmussen [2005], we consider the binary version by considering 3's vs 5's. We use 767 data points for training and rest 763 for testing. The results shown in Fig. 4.15 show the same trend as the ionosphere dataset.

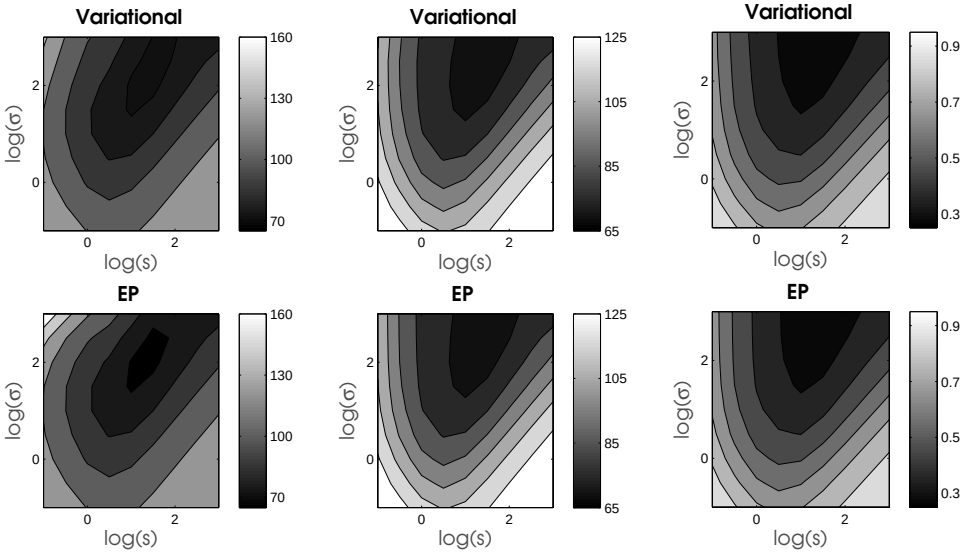
The advantage of our approach is that, unlike EP, it does not suffer from any numerical issues (for example, no negative variances) and is guaranteed to converge. The computational complexity of our approach is almost identical to that of EP. For example, compare Algorithm 3 to the algorithm for EP given in Rasmussen and Williams [2006] (see Algorithm 3.5 in the book). Both our algorithm and EP run scalar updates, followed by a rank one update of posterior covariance $\mathbf{V}$. An additional advantage of our ap-



(a) $\boldsymbol{\theta} = \{-1, -1\}$ (b) $\theta = \{-1, 4\}$

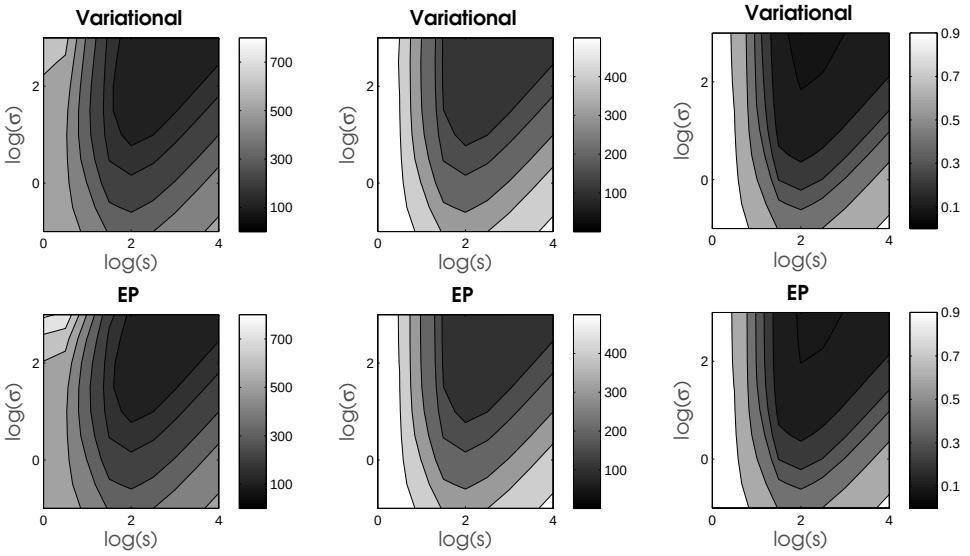
Figure 4.13: Comparison of approximate posterior for two parameter settings, shown at the bottom of the plot with $\boldsymbol{\theta} = \{\log(s), \log(\sigma)\}$. We plot elements of the mean and covariance obtained with the variational approach vs those obtained with EP.

proach is that it extends easily to models such as factor analysis, there too with guaranteed convergence. For this model class, EP usually leads to non-standard and usually non-monotonic optimization since the inference and learning steps do not optimize the same lower bound (as we discussed in Section 2.3.3). Finally, our approach easily extends to other data types such as categorical and mixed-data, as we show in next two chapters.



(a) Negative ELBO (b) Negative EPapprox (c) Prediction error

Figure 4.14: EP vs variational on the ionosphere dataset.



(a) Negative ELBO (b) Negative EPapprox (c) Prediction error

Figure 4.15: EP vs variational on the 'USPS-3vs5' dataset.

Chapter 5

Variational Learning of Categorical LGMs

In this chapter, we discuss variational learning for categorical LGMs. Similarly to the binary case, the variational learning is intractable due to non-conjugacy of the Gaussian prior to categorical data likelihoods, such as multinomial logit/probit. Existing methods for tractable variational learning are inaccurate and slow.

We make two contributions in this regard. Our first contribution is the application of the Bohning bound for the multinomial logit likelihood. The Bohning bound leads to a fast variational learning algorithm, but can be inaccurate at times. We present a theoretical comparison of existing LVBs, discussing conditions under which they are reasonably accurate. Unfortunately, all of the existing LVBs for the multinomial logit likelihood can be inaccurate at times, and designing LVBs with error guarantees remains a difficult task. We take a different approach to solve this problem. We propose a new likelihood, called stick breaking likelihood, for categorical LGMs. The main advantage of this likelihood is the availability of accurate LVBs with error guarantees, leading to an accurate variational learning. With application to real datasets, we show that the variational learning with the proposed likelihood is more accurate than variational learning with existing likelihoods.

5.1 Categorical LGM

We start by defining categorical LGM to model categorical data vectors $\mathbf{y}_n$. Each element y_{dn} of $\mathbf{y}_n$ takes values from a finite discrete set $S_d = \{C_0, C_1, C_2, \ldots, C_{K_d}\}$, where C_k is the k'th category. For simplicity, we assume that $K_d = K$ for all d. We use a dummy encoding for y_{dn}, that is, we encode it as a binary vector $\mathbf{y}_{dn}$ of length $K+1$ where we set $\mathbf{y}_{kdn}$ to 1 if $y_{dn} = C_{k+1}$.

Similarly to other LGMs, the latent variables follow the Gaussian distri-

bution as shown in Eq. 5.1. The probability of each categorical observation $\mathbf{y}_{dn}$ is parameterized in terms of the linear predictor $\boldsymbol{\eta}_{dn} = \mathbf{W}_d\mathbf{z}_n + \mathbf{w}_{0d}$ of the latent variables $\mathbf{z}_n$ as seen in Eqs. 5.2. The k'th element of $\boldsymbol{\eta}_{dn}$ is predictor for the k'th category. The matrix $\mathbf{W}_d$ is the factor loading matrix and the vector $\mathbf{w}_{0d}$ is the offset vector, both taking real values. Given $\boldsymbol{\eta}_{dn}$, the data vector is modeled using a likelihood $p(y_{dn}|\boldsymbol{\eta}_{dn})$, as shown in Eq. 5.3 We will discuss the exact form of the likelihood in the next section.

$$p(\mathbf{z}_n|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{5.1}$$

$$\boldsymbol{\eta}_{dn} = \mathbf{W}_d\mathbf{z}_n + \mathbf{w}_{0d} \tag{5.2}$$

$$p(\mathbf{y}_n|\boldsymbol{\eta}_n) = \prod_{d=1}^{D} p(\mathbf{y}_{dn}|\boldsymbol{\eta}_{dn}) \tag{5.3}$$

The parameter set $\boldsymbol{\theta}$ is the set of parameters required to define $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{W}, \mathbf{w}_0\}$ where $\mathbf{W}$ and $\mathbf{w}_0$ are the matrix containing $\mathbf{W}_d$ and $\mathbf{w}_{0d}$ as rows.

5.2 Multinomial Logit Likelihood

We discussed the multinomial logit likelihood in Section 1.3.3. This likelihood can be expressed in terms of the log-sum-exp (LSE) function, $\text{lse}(\boldsymbol{\eta}) = \log \sum_j \exp(\eta_j)$, as shown below,

$$p(y = C_k|\boldsymbol{\eta}) = \frac{e^{\eta_k}}{\sum_{j=0}^{K} e^{\eta_j}} = \exp\left[\mathbf{y}^T\boldsymbol{\eta} - \text{lse}(\boldsymbol{\eta})\right] \tag{5.4}$$

where $\mathbf{y}$ is the dummy encoding of observation $y = C_k$. To make the model identifiable, sometimes we set an element of $\boldsymbol{\eta}$ to 0, say $\eta_0 = 0$; see Section 1.3.3 for a detailed discussion of this.

The evidence lower bound to the marginal likelihood is intractable for the multinomial logit likelihood due to the LSE term. This can be seen below from the expression for the expectation of log-likelihood.

$$\mathbb{E}_{q(\boldsymbol{\eta}|\widetilde{\boldsymbol{\gamma}})}[\log p(y|\boldsymbol{\eta})] = \mathbf{y}^T\widetilde{\mathbf{m}} - \mathbb{E}_{q(\boldsymbol{\eta}|\widetilde{\boldsymbol{\gamma}})}[\text{lse}(\boldsymbol{\eta})] \tag{5.5}$$

The expectation is with respect to the approximate Gaussian distribution, $q(\boldsymbol{\eta}|\widetilde{\boldsymbol{\gamma}}) = \mathcal{N}(\boldsymbol{\eta}|\widetilde{\mathbf{m}}, \widetilde{\mathbf{V}})$. To obtain tractable lower bounds, we find an upper bound to the LSE term such that the expectation of the upper bound is tractable. We describe few LVBs obtained with this approach, before discussing our contributions to the variational learning. Note that, all LVBs discussed are jointly concave with respect to $\widetilde{\boldsymbol{\gamma}}$.

5.3 Existing LVBs for Multinomial Logit Likelihood

In this section, we review existing LVBs for the multinomial logit likelihood. Unfortunately, none of the bounds are accurate all the time and, unlike the Bernoulli logit likelihood, it is difficult to design bounds with bounded error. The main source of error in most of the bounds is the local nature of the approximation. Most of the bounds are derived using the delta approximation of the LSE function, which involves approximating the expectation using Taylor's expansion at the mean. The LSE function is highly skewed function and being accurate at the mean does not ensure global tightness. Another problem with these bounds is that they do not depend on the off-diagonal elements of $\widetilde{\mathbf{V}}$, and can be inaccurate when off-diagonal elements are significantly large, for example, when the latent variables are highly correlated. We will prove a theoretical result in Section 5.5 regarding this.

The log bound

The most popular bound is the log bound, proposed by Blei and Lafferty [2006], and is shown below,

$$\underline{f}^L(\boldsymbol{\theta}, \widetilde{\boldsymbol{\gamma}}) := \mathbf{y}^T \widetilde{\mathbf{m}} - \text{lse}(\widetilde{\mathbf{m}} + \widetilde{\mathbf{v}}/2) \tag{5.6}$$

with $\widetilde{\mathbf{v}}$ being the diagonal of $\widetilde{\mathbf{V}}$. This bound can be derived using the zero-order delta method; see Appendix A.6 for derivation. The advantage of the log bound is that it is a very simple bound. The gradients of the log bound are shown below,

$$\mathbf{g}^m = \mathbf{y} - \mathbf{t}, \quad \mathbf{g}^v = -\tfrac{1}{2}\text{diag}(\mathbf{t}), \quad \mathbf{t} := e^{(\widetilde{\mathbf{m}}+\widetilde{\mathbf{v}}/2)-\text{lse}(\widetilde{\mathbf{m}}+\widetilde{\mathbf{v}}/2)} \tag{5.7}$$

Since it does not have local variational parameters and since the gradients take a very simple form, its implementation is easy.

The tilted bound

The tilted bound was recently proposed by Knowles and Minka [2011] and is shown below,

$$\underline{f}^T(\boldsymbol{\theta}, \widetilde{\boldsymbol{\gamma}}, \mathbf{a}) := \mathbf{y}^T \widetilde{\mathbf{m}} - \tfrac{1}{2}\sum_{k=0}^{K} a_k^2 \widetilde{v}_j - \log \sum_{k=0}^{K} e^{\widetilde{m}_k + (1-2a_k)\widetilde{v}_k/2} \tag{5.8}$$

where $a_k \in [0, 1]$. Detailed derivation is given in Appendix A.7. This bound reduces to the log bound for $a_k = 0$ and hence is a generalization of the log bound. Updates of $\mathbf{a}$ can be done using an iterated procedure, and hence computation overhead is not much higher than the log bound; see [Knowles and Minka, 2011] for details. Similar to the log bound, the tilted bound is also based on the delta approximation at the mean and may be only tight locally (see derivation in Appendix A.7). In addition, this bound also does not involve the off-diagonal elements of $\widetilde{\mathbf{V}}$ and may not perform well when those elements are significant. The bound, however, is shown to perform well by the authors when $\widetilde{\mathbf{V}}$ is diagonal. A variant of this bound which includes the off-diagonal elements of $\mathbf{V}$ can also be designed, although we are not aware of any work on this variant. See Appendix A.7 for the details of the variant. We leave comparison with this variant as future work.

The product of sigmoid bound

The product of sigmoid (POS) bound, proposed by Bouchard [2007], is given as follows,

$$\underline{f}^S(\boldsymbol{\theta}, \widetilde{\boldsymbol{\gamma}}, \beta) := \mathbf{y}^T \widetilde{\mathbf{m}} - \beta - \sum_{k=0}^{K} \mathbb{E}_{q(\eta|\widetilde{\boldsymbol{\gamma}}_k)}[\mathrm{llp}(\eta - \beta)] \tag{5.9}$$

where $\beta \in \mathbb{R}$. The POS bound can be derived using the following inequality,

$$\prod_{k=1}^{K}(1 + e^{\eta_k - \beta}) \geq \sum_{k=1}^{K} e^{\eta_k - \beta} = e^{-\beta} \sum_{k=1}^{K} e^{\eta_k} \tag{5.10}$$

Taking log on both sides and rearranging, we get the following upper bound on the LSE function: $\mathrm{lse}(\boldsymbol{\eta}) \leq \beta + \sum_{k=1}^{K} \log(1 + e^{\eta_k - \beta})$. We substitute this Eq. 5.5 to get the POS bound of Eq. 5.9. The advantage of this bound is that it is expressed in terms of the LLP function for which many bounds exist. Bouchard [2007] uses the Jaakkola bound [Jaakkola and Jordan, 1996] which can be inaccurate at times as we showed in Chapter 4. However, use of the Jaakkola bound leads to closed form updates in EM algorithm, which is useful. A more accurate version can be designed by using the piecewise bound for the LLP function, but will be slower since it requires gradient methods to be used. The LLP function asymptotically approaches the LSE function in the direction of a predictor, and hence the POS function is expected to be accurate in that direction. However, it is not accurate in general as reported by Knowles and Minka [2011].

First-order delta method

The Delta method is used to approximate moments of a function using the Taylor expansion [Casella and Berger, 2001]. The zeroth-order delta method can be used to derive the log bound as shown in Appendix A.6. In this section, we describe an approximation based on the first-order delta method. A first-order approximation to the expectation of a function f is obtained by taking expectation of a first-order Taylor expansion around $\widetilde{\mathbf{m}}$, as shown below. Here, $\mathbf{H}_f$ is the Hessian of f.

$$\begin{aligned}
&\mathbb{E}_{q(\boldsymbol{\eta}|\widetilde{\boldsymbol{\gamma}})}[f(\boldsymbol{\eta})] \\
&\approx \mathbb{E}_{q(\boldsymbol{\eta}|\widetilde{\boldsymbol{\gamma}})}[f(\widetilde{\mathbf{m}}) + (\boldsymbol{\eta} - \widetilde{\mathbf{m}})^T \mathbf{g}_f(\widetilde{\mathbf{m}}) + \tfrac{1}{2}(\boldsymbol{\eta} - \widetilde{\mathbf{m}})^T \mathbf{H}_f(\mathbf{m})(\boldsymbol{\eta} - \widetilde{\mathbf{m}})] \quad (5.11) \\
&= f(\widetilde{\mathbf{m}}) + \tfrac{1}{2}\mathrm{Tr}\left[\mathbf{H}_f(\mathbf{m})\widetilde{\mathbf{V}}\right] \quad (5.12)
\end{aligned}$$

We choose f to be the LSE function, apply the first-order approximation to $\mathbb{E}_{q(\boldsymbol{\eta}|\widetilde{\boldsymbol{\gamma}})}[\mathrm{lse}(\boldsymbol{\eta})]$, and substitute in Eq. 5.5 to the following approximation,

$$\underline{f}^D(\boldsymbol{\theta}, \widetilde{\boldsymbol{\gamma}}, \tilde{\boldsymbol{\beta}}) := \mathbf{y}^T \widetilde{\mathbf{m}} - \mathrm{lse}(\widetilde{\mathbf{m}}) - \tfrac{1}{2}\mathrm{Tr}\left[\mathbf{H}_{lse}(\widetilde{\mathbf{m}})\widetilde{\mathbf{V}}\right] \quad (5.13)$$

This approximation has been applied to some LGMs, such as discrete choice models Braun and McAuliffe [2010] and correlated topic model by Ahmed and Xing [2007]. However, since this is not a lower bound, we loose monotonicity of the EM algorithm and diagnosing the convergence becomes difficult. A more serious problem however is that the approximation is accurate only locally, just like the log bound. However, it is expected to be more accurate than the log bound, since it is based on the first-order approximation. We will show this in section 5.5.

5.4 A New LVB: The Bohning Bound

The Bohning bound, a quadratic bound discussed in Chapter 4 for Bernoulli logit likelihood, can be generalized to the multinomial logit likelihood. It is simple to define the Bohning bound for the case when one of the entry of $\boldsymbol{\eta}$ is set to 0. This assumption can be easily relaxed, as described later in Section 5.4.1. For now, let us assume $\eta_0 = 0$ and redefine $\boldsymbol{\eta}$ to be a vector of rest of the elements, i.e. $\boldsymbol{\eta} = (\eta_1, \eta_2, \ldots, \eta_K)$. We redefine $\mathbf{y}$ and the Gaussian distribution $q(\boldsymbol{\eta}|\widetilde{\boldsymbol{\gamma}})$ accordingly. Given this, the Bohning bound is

defined as following,

$$\underline{f}^B(\mathbf{y}, \widetilde{\boldsymbol{\gamma}}, \boldsymbol{\psi}) := \mathbf{y}^T \widetilde{\mathbf{m}} - \tfrac{1}{2}\widetilde{\mathbf{m}}^T \mathbf{A}\widetilde{\mathbf{m}} + \mathbf{b}_{\boldsymbol{\psi}}^T \widetilde{\mathbf{m}} - c_{\boldsymbol{\psi}} - \tfrac{1}{2}\mathrm{Tr}\left(\mathbf{A}\widetilde{\mathbf{V}}\right) \tag{5.14}$$

$$\mathbf{A} := \tfrac{1}{2}[\mathbf{I}_K - \mathbf{1}_K \mathbf{1}_K^T/(K+1)] \tag{5.15}$$

$$\mathbf{b}_{\boldsymbol{\psi}} := \mathbf{A}\boldsymbol{\psi} - \mathbf{g}_{\boldsymbol{\psi}} \tag{5.16}$$

$$c_{\boldsymbol{\psi}} := \tfrac{1}{2}\boldsymbol{\psi}^T \mathbf{A}\boldsymbol{\psi} - \mathbf{g}_{\boldsymbol{\psi}}^T \boldsymbol{\psi} + \mathrm{lse1}(\boldsymbol{\psi}) \tag{5.17}$$

$$\mathbf{g}_{\boldsymbol{\psi}} := \exp[\boldsymbol{\psi} - \mathrm{lse1}(\boldsymbol{\psi})] \tag{5.18}$$

where $\mathbf{I}_K$ is the identity matrix of size K, $\mathbf{1}_K$ is a K-length vector of ones, $\boldsymbol{\psi} \in \mathbb{R}^K$ is the local variational parameter vector, and $\mathrm{lse1}(\mathbf{x}) := \log[1 + \sum_{k=1}^K \exp(x_k)]$.

For the binary case, this bound reduces to the bound discussed earlier in Chapter 4 for the Bernoulli logit likelihood. The Bohning bound for the multinomial logit likelihood shares all the properties of the Bohning bound for the Bernoulli logit likelihood. For example, just like its binary counterpart, this bound has a fixed curvature parameter $\mathbf{A}$ which allows us to design a fast algorithm.

Similar to the binary case, the Bohning bound is maximized when $\boldsymbol{\psi}_* = \widetilde{\mathbf{m}}$. Proof is exactly same as the binary case; see Section 4.4.2. Substituting this in the Bohning bound of Eq. 5.14, we can simplify the Bohning bound. We simply substitute the value of $\boldsymbol{\psi}_*$ in $\mathbf{b}_\psi$ and c_ψ, and simplify to obtain the expression below.

$$\underline{f}^B(\mathbf{y}, \widetilde{\boldsymbol{\gamma}}, \boldsymbol{\psi}_*) = \mathbf{y}^T \widetilde{\mathbf{m}} - \mathrm{lse}(\widetilde{\mathbf{m}}) - \tfrac{1}{2}\mathrm{Tr}(\mathbf{A}\widetilde{\mathbf{V}}) \tag{5.19}$$

We now give a detailed derivation of the Bohning bound and then describe the learning algorithm.

5.4.1 Derivation

We derive the Bohning bound using a Taylor series expansion of $\mathrm{lse1}(\boldsymbol{\eta})$ around a point $\boldsymbol{\psi} \in \mathbb{R}^K$ as shown in Eq. 5.20. Here, $\mathbf{g}_{\boldsymbol{\psi}}$ and $\mathbf{H}_{\boldsymbol{\psi}}$ are the gradient and Hessian respectively as defined in Eq. 5.22 and 5.23, and the equality holds for some $\boldsymbol{\chi} \in \mathbb{R}^K$ due to Taylor's theorem Rudin [2006]. An upper bound to $\mathrm{lse1}(\boldsymbol{\eta})$ is found by replacing the second derivative term by an upper bound. It can be shown that $\mathbf{x}^T \mathbf{H}_{\boldsymbol{\chi}} \mathbf{x} \leq \mathbf{x}^T \mathbf{A}\mathbf{x}$ for all $\mathbf{x}$ and $\boldsymbol{\chi}$, where $\mathbf{A} := [\mathbf{I}_K - \mathbf{1}_K \mathbf{1}_K^T/(K+1)]/2$ (see Bohning [1992] for a proof). Using

this we get the upper bound shown in Eq. 5.21.

$$\text{lse1}(\boldsymbol{\eta}) = \text{lse1}(\boldsymbol{\psi}) + (\boldsymbol{\eta} - \boldsymbol{\psi})^T \mathbf{g}_{\boldsymbol{\psi}} + \frac{1}{2}(\boldsymbol{\eta} - \boldsymbol{\psi})^T \mathbf{H}_{\boldsymbol{\chi}} (\boldsymbol{\eta} - \boldsymbol{\psi}) \tag{5.20}$$

$$\leq \text{lse1}(\boldsymbol{\psi}) + (\boldsymbol{\eta} - \boldsymbol{\psi})^T \mathbf{g}_{\boldsymbol{\psi}} + \frac{1}{2}(\boldsymbol{\eta} - \boldsymbol{\psi})^T \mathbf{A} (\boldsymbol{\eta} - \boldsymbol{\psi}) \tag{5.21}$$

$$\mathbf{g}_{\boldsymbol{\psi}} := \exp[\boldsymbol{\psi} - \text{lse1}(\boldsymbol{\psi})] \tag{5.22}$$

$$\mathbf{H}_{\boldsymbol{\psi}} := \text{diag}(\mathbf{g}_{\boldsymbol{\psi}}) - \mathbf{g}_{\boldsymbol{\psi}} \mathbf{g}_{\boldsymbol{\psi}}^T \tag{5.23}$$

We substitute this in Eq. 5.5 and rearrange terms to get the Bohning bound shown in Eq. 5.14.

The assumption that $\eta_0 = 0$ can be relaxed easily by redefining $\mathbf{A}$ to be $[\mathbf{I}_{K+1} - \mathbf{1}_{K+1}\mathbf{1}_{K+1}^T/(K+1)]$ and by doing Taylor expansion of $\text{lse}(\boldsymbol{\eta})$ instead of $\text{lse1}(\boldsymbol{\eta})$. However, the new $\mathbf{A}$ is a positive semi-definite matrix and might give rise to numerical problems. Another way to relax the condition is to rewrite log of the multinomial logit likelihood as $\log p(y = C_k|\boldsymbol{\eta}) = \eta_k - \log \sum_k \exp(\eta_k - \eta_0)$, such that the first element in the summation is 0, and then apply Taylor's expansion on $\text{lse1}(\mathbf{x})$ to get a bound.

5.4.2 Variational Learning

For a tractable lower bound, we need to bound $\mathbb{E}_{q(\eta|\widetilde{\gamma}_{dn})}[\log p(\mathbf{y}_{dn}|\eta)]$ for the (d,n)'th observation. The Bohning bound to this term is shown below, where $\mathbf{b}_{\psi,dn}$ and $c_{\psi,dn}$ are functions of the local variational parameter ψ_{dn} and are defined as in Eq. 5.16 and 5.17.

$$\underline{f}^B(\mathbf{y}_{dn}, \widetilde{\boldsymbol{\gamma}}_{dn}, \boldsymbol{\psi}_{dn}) := \mathbf{y}_{dn}^T \widetilde{\mathbf{m}}_{dn} - \tfrac{1}{2}\widetilde{\mathbf{m}}_{dn}^T \mathbf{A} \widetilde{\mathbf{m}}_{dn} + \mathbf{b}_{\boldsymbol{\psi},dn}^T \widetilde{\mathbf{m}}_{dn} - c_{\boldsymbol{\psi},dn} - \tfrac{1}{2}\text{Tr}\left(\mathbf{A}\widetilde{\mathbf{V}}_{dn}\right) \tag{5.24}$$

To compute gradients with respect to $\boldsymbol{\gamma}_n$ and $\boldsymbol{\theta}$, we need gradients with respect to $\widetilde{\mathbf{m}}_{dn}$ and $\widetilde{\mathbf{v}}_{dn}$ (see Section 3.5.1), which are given below,

$$\mathbf{g}_{dn}^m = (\mathbf{y}_{dn} + \mathbf{b}_{\psi,dn}) - \mathbf{A}\widetilde{\mathbf{m}}_{dn} \quad , \quad \mathbf{G}_{dn}^v = -\tfrac{1}{2}\mathbf{A} \tag{5.25}$$

We now derive updates for $\boldsymbol{\gamma}_n$, $\boldsymbol{\theta}$, and $\boldsymbol{\psi}_{dn}$.

Update of variables parameter $\boldsymbol{\psi}_{dn}$ remains same as before, $\boldsymbol{\psi}_{dn} = \widetilde{\mathbf{m}}_{dn}$. For updates of $\boldsymbol{\gamma}_n$, $\boldsymbol{\theta}$, we substitute the expressions for $\mathbf{g}_{dn}^m$ and $\mathbf{G}_{dn}^v$ in the generalized gradient expressions given in Algorithm 1. We set the gradients to zero and simplify to get the updates below. Details are given in Appendix A.4.

The E-step updates are shown below,

$$\mathbf{V} = \left(\boldsymbol{\Sigma}^{-1} + \mathbf{W}^T \bar{\mathbf{A}} \mathbf{W}\right)^{-1} \tag{5.26}$$

$$\mathbf{m}_n = \mathbf{V}\left[\mathbf{W}^T(\mathbf{y}_n + \mathbf{b}_n - \bar{\mathbf{A}}\mathbf{w}_0) + \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right] \tag{5.27}$$

with $\mathbf{b}_n$ being the vector of $b_{\psi,dn}, \forall d$ and $\bar{\mathbf{A}}$ is a block diagonal matrix of size $DK \times DK$ with each block equal to $\mathbf{A}$.

The M-step updates are as follows,

$$\mathbf{W} = \left[\sum_{n=1}^{N} \left\{\bar{\mathbf{A}}^{-1}(\mathbf{y}_n + \mathbf{b}_n) - \mathbf{w}_0\right\} \mathbf{m}_n^T\right] \left[\sum_{n=1}^{N} \mathbf{V} + \mathbf{m}_n \mathbf{m}_n^T\right]^{-1} \tag{5.28}$$

$$\mathbf{w}_0 = \frac{1}{N} \sum_{n=1}^{N} \bar{\mathbf{A}}^{-1}(\mathbf{y}_n + \mathbf{b}_n) - \mathbf{W}\mathbf{m}_n \tag{5.29}$$

along with updates of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ which remains same as Eq. 3.37.

Similar to the binary case, the Bohning bound leads to simple closed form updates with posterior covariance $\mathbf{V}$ being independent of the data. The updates above are exactly same as those of Gaussian LGMs, but now, with data vectors $\bar{\mathbf{A}}^{-1}(\mathbf{y} + \mathbf{b}_n)$ and noise covariance matrix $\bar{\mathbf{A}}^{-1}$. The lower bound for the (d, n)'th measurement is an unnormalized multivariate Gaussian such that $p(\mathbf{y}_{dn}|\boldsymbol{\eta}_{dn}) \geq Z_\psi \mathcal{N}(\tilde{\mathbf{y}}_{dn}|\boldsymbol{\eta}_{dn}, \mathbf{A}^{-1})$ for some Z_ψ with $\tilde{\mathbf{y}}_{dn} = \mathbf{A}^{-1}(\mathbf{y}_{dn} + \mathbf{b}_{\psi,dn})$.

The variational algorithm takes the same form as Algorithm 5. Here again, since $\mathbf{V}$ is same for all n, we only need to compute it once during the E-step, instead of computing it for all n separately. This simplification leads to a huge computational saving since computation of $\mathbf{V}$ involves inversion of a square matrix of size L, making complexity of this step $O(L^3 + DK^2L^2)$, which is independent of n. Computing $\mathbf{m}_n$ requires a multiplication of two matrices of sizes $L \times DK$ and $DK \times L$, plus N multiplications of a $L \times DK$ matrix with a $DK \times 1$ vector for few iterations. This makes the total cost of the E-step to be $O(L^3 + DK^2L^2 + NDKLI)$, where I is the number of iterations taken for convergence in the E-step. The cost of M-step is same as the Gaussian LGM which is $O(L^3 + NL^2 + NDKL)$. Also, the memory cost is same as well, which is $O(L^2 + L\min(N, DK))$.

5.5 Error Analysis

In this section, we theoretically analyze the relative errors between different LVBs. Our goal here is to show that no single bound is accurate all the

time. However, the analysis reveals that, in terms of accuracy, the bounds can be roughly ranked as the following sequence of increasing accuracy,

$$\text{Bohning, Log, Tilted, First-order Delta} \tag{5.30}$$

By "roughly", we mean that most of the time the bound on the left is less accurate than the bound on the right. The trends in speed are almost reversed, i.e. the bound in the left leads to the fastest algorithm. We did not include the POS bound here because it is difficult to theoretically compare with other bound since the POS bound takes a very different form than other bounds.

We start our analysis with the log and Bohning bound. The log bound is more accurate than the Bohning bound in general, but not always. The following theorem bounds the difference between the two bounds (see the proof in Appendix A.8).

Theorem 5.5.1. *For all $\mathbf{y}, \widetilde{\boldsymbol{\gamma}}$ and $\boldsymbol{\psi}$, the difference between the log and Bohning bound, denoted by $\Delta(\mathbf{y}, \widetilde{\boldsymbol{\gamma}}, \boldsymbol{\psi}) := \underline{f}^L(\mathbf{y}, \widetilde{\boldsymbol{\gamma}}) - \underline{f}^B(\mathbf{y}, \widetilde{\boldsymbol{\gamma}}, \boldsymbol{\psi})$, is bounded as follows,*

$$\tfrac{1}{2}\left[Tr(\mathbf{A}\widetilde{\mathbf{V}}) - \widetilde{v}_{max}\right] \leq \Delta(\mathbf{y}, \widetilde{\boldsymbol{\gamma}}, \boldsymbol{\psi}) \leq \tfrac{1}{2}\left[Tr(\mathbf{A}\widetilde{\mathbf{V}}) - \widetilde{v}_{min}\right] \tag{5.31}$$

where $\widetilde{v}_{max}, \widetilde{v}_{min}$ are the maximum and minimum diagonal elements of $\widetilde{\mathbf{V}}$ and $\mathbf{A}$ is the curvature matrix for the Bohning bound defined in Eq. 5.15.

The condition 5.31 can be used to theoretically compare the two bounds. When the lower bound in Eq. 5.31 is greater than 0, the log bound will be better than the Bohning bound. Similarly, when the upper bound is less than 0, then the Bohning bound is better. Note that the trace term $\text{Tr}(\mathbf{A}\widetilde{\mathbf{V}})$ is always greater than or equal to 0, since both $\mathbf{A}$ and $\widetilde{\mathbf{V}}$ are positive semi-definite. So Δ is bounded between $\widetilde{v}_{\max}$ and $\widetilde{v}_{min}$ around a non-negative $\text{Tr}(\mathbf{A}\widetilde{\mathbf{V}})$. The interplay between the off-diagonal and the diagonal elements decides whether one bound is better than the other.

We can construct examples where one bound will be better than the other. The trace term can be expanded as follows,

$$\text{Tr}(\mathbf{A}\widetilde{\mathbf{V}}) = \tfrac{1}{2}\text{Tr}\left\{\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{K+1}\right]\widetilde{\mathbf{V}}\right\} = \tfrac{1}{2}\sum_{k=0}^{K}\widetilde{\mathbf{V}}_{kk} - \tfrac{1}{2}\sum_{i=0}^{K}\sum_{j=0}^{K}\frac{\widetilde{\mathbf{V}}_{ij}}{(K+1)} \tag{5.32}$$

This involves difference between the sum of the diagonal and the sum of the full matrix weighted by $1/(K+1)$. Consider a case where $\mathbf{V}$ is a diagonal

matrix with each diagonal element equal to v. Then $\mathrm{Tr}(\mathbf{A}\widetilde{\mathbf{V}})/2 = Kv/2$, which is always greater than $v_{\max} = v$, making the left hand size of inequality 5.31 greater than 0. Hence, for this case, the log bound will be more accurate than the Bohning bound. Now, consider a 3×3 matrix $\widetilde{\mathbf{V}}$ with all diagonal entries set to v, and all off-diagonal entries equal to zero, except $\widetilde{V}_{12}$ and $\widetilde{V}_{21}$ which we set to ϵ with $0 < \epsilon < v$. The new matrix $\widetilde{\mathbf{V}}$ is positive definite. Now, $\mathrm{Tr}(\mathbf{A}\widetilde{\mathbf{V}}) = v - \epsilon/3$ which is always less than $v_{min} = v$, making the right hand side negative. Hence, the Bohning bound will be better than the log bound for this case.

Note that, in Eq. 5.32, the contribution of the off-diagonal elements decreases as K increases. So the size of the set of $\widetilde{\mathbf{V}}$, for which the Bohning bound is better than the log bound, decreases as K increases. This theorem however brings out an important point that the log bound can be inaccurate for cases where the off-diagonal elements of $\mathbf{V}$ are significant. The same applies to all the other bounds. Hence, we can conclude that, most of the times, the Bohning is less accurate than the log bound.

The tilted bound will always be more accurate than the log bound, since the log bound is a special case of the tilted bound with $a_k = 0, \forall k$.

Comparison with the first-order delta approximation is complicated since it is not a bound but an approximation. However, we can bound the extent to which the delta approximation will vary around each bound. For example, the delta approximation always has a higher value than the Bohning bound. This can be proved by looking at the difference between the delta method and the Bohning bound, $\Delta \geq \frac{1}{2}\mathrm{Tr}[\mathbf{A} - \mathbf{H}_{lse}(\widetilde{\mathbf{m}})\widetilde{\mathbf{V}}] \geq 0$, since $\mathbf{A} - \mathbf{H}_{lse}$ is a positive definite matrix. This however does not prove that the delta method will be a better approximation since it is not a bound. Similarly, the difference between the log bound and the first-order delta approximation is bounded as follows,

$$\tfrac{1}{2}\left[\widetilde{v}_{\min} - \mathrm{Tr}[\mathbf{H}_{lse}(\widetilde{\mathbf{m}})\widetilde{\mathbf{V}}]\right] \leq \Delta \leq \tfrac{1}{2}\left[\widetilde{v}_{max} - \mathrm{Tr}[\mathbf{H}_{lse}(\widetilde{\mathbf{m}})\widetilde{\mathbf{V}}]\right] \tag{5.33}$$

Since the trace term $\mathrm{Tr}[\mathbf{H}_{lse}(\widetilde{\mathbf{m}})\widetilde{\mathbf{V}}]$ is always positive, the difference between the two bounds varies around a negative value. Hence, we can conclude that most of the times the first-order delta approximation will take higher values than the log bound. This also proves that the delta approximation, despite being an approximation, is never going to be higher than the true log marginal likelihood by $v_{\max}/2$.

5.6 Stick Breaking Likelihood

Variational learning with the multinomial logit likelihood can be inaccurate due to error incurred in the LVBs. We propose an alternative generalization of the logit function, for which accurate LVBs can be designed. We refer to this as the *stick breaking* likelihood. We show that variational learning for our proposed likelihood can be more accurate than that of the multinomial logit likelihood.

In the stick breaking parameterization, we use a logit function to model the probability of the first category as $\sigma(\eta_0)$ where $\sigma(x) = 1/(1 + \exp(x))$. This is the first piece of the stick. The length of the remainder of the stick is $(1 - \sigma(\eta_0))$. We can model the probability of the second category as a fraction $\sigma(\eta_1)$ of the remainder of the stick left after removing $\sigma(\eta_0)$. We can continue in this way until we have defined all the stick lengths up to K. The last category then receives the remaining stick length, as seen below.

$$\begin{aligned} p(y = C_0|\boldsymbol{\eta}) &= \sigma(\eta_0) \\ p(y = C_k|\boldsymbol{\eta}) &= \prod_{j \leq k-1} (1 - \sigma(\eta_j))\sigma(\eta_k), 0 < k < K \\ p(y = C_K|\boldsymbol{\eta}) &= \prod_{j=1}^{K-1} (1 - \sigma(\eta_j)) \end{aligned} \tag{5.34}$$

An example of this construction for 4 categories is shown in Figure 5.1. The probabilities (stick lengths) are all positive and sum to one; they thus define a valid probability distribution. We can also use a different function for $\sigma(x)$ such as the probit function, but we use the logit function since it allows us to use efficient variational bounds. The stick-breaking parameterization can be written more compactly as shown in Eq. 5.35.

$$p(y = C_k|\boldsymbol{\eta}) = \exp[\eta_k - \sum_{j \leq k} \log(1 + e^{\eta_j})] \tag{5.35}$$

Models for multinomial probit/logit regression have wide coverage in the statistics and psychology literature. Both the multinomial-probit and multinomial-logit links are used extensively [Albert and Chib, 1993; Holmes and Held, 2006]. These link functions do not assume any ordering of categories and it is understood that these parameterizations give similar performance and qualitative conclusions. On the other hand, learning with these link functions is difficult.

Our stick-breaking construction simplifies the learning by constructing a categorical likelihood using simpler binary likelihood functions as shown

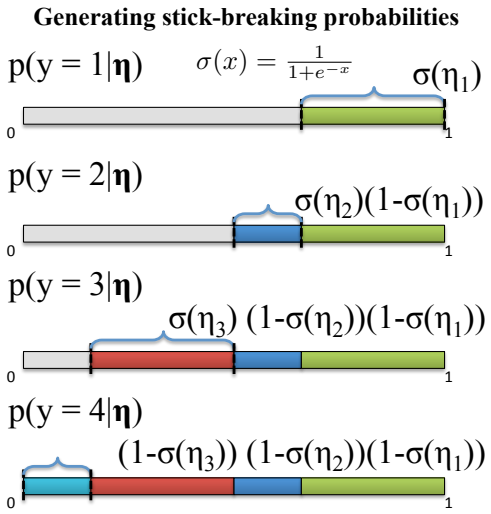


Figure 5.1: Stick-breaking likelihood for 4 categories. We start with a stick of length 1, and break it at $\sigma(\eta_1)$ to get the probability of first category. We then split rest of the stick at $\sigma(\eta_2)$ to get the probability of second category. We continue this until the last category, probability of which is equal to whatever is left of the stick.

in Eq. 5.34. Each η_k can be interpreted as the log-ratio: $\eta_k = \log[p(y = C_k|\boldsymbol{\eta})/p(y > C_k|\boldsymbol{\eta})]$. This implies that, given a particular ordering of categories, each η_k defines a decision boundary in the latent space $\mathbf{z}$, that separates the k'th category from all categories $j > k$. If such a separation is difficult to attain given an ordering of categories, the stick-breaking likelihood may not give good predictions. In practice, such separability is easier to achieve in latent variable models such as Gaussian process and factor analysis. Our results on real-world datasets confirm this. An illustration is shown in Figure 5.2.

This interpretation also relates the stick-breaking likelihood to the continuation ratio model discussed in Section 1.3.4 for ordinal data. For ordinal data, the predictor is a scalar. The stick-breaking likelihood is simply a generalization of the continuation ratio model to the categorical case. See Kim [2002] for a stick-breaking interpretation of the continuation ratio model.

The stick-breaking parameterization also has important advantages over the multinomial-logit model in terms of variational approximations. As we saw in previous sections, the multinomial-logit parameterization requires bounding the $\text{lse}(\boldsymbol{\eta})$ function. It is not known how to obtain tight bounds on

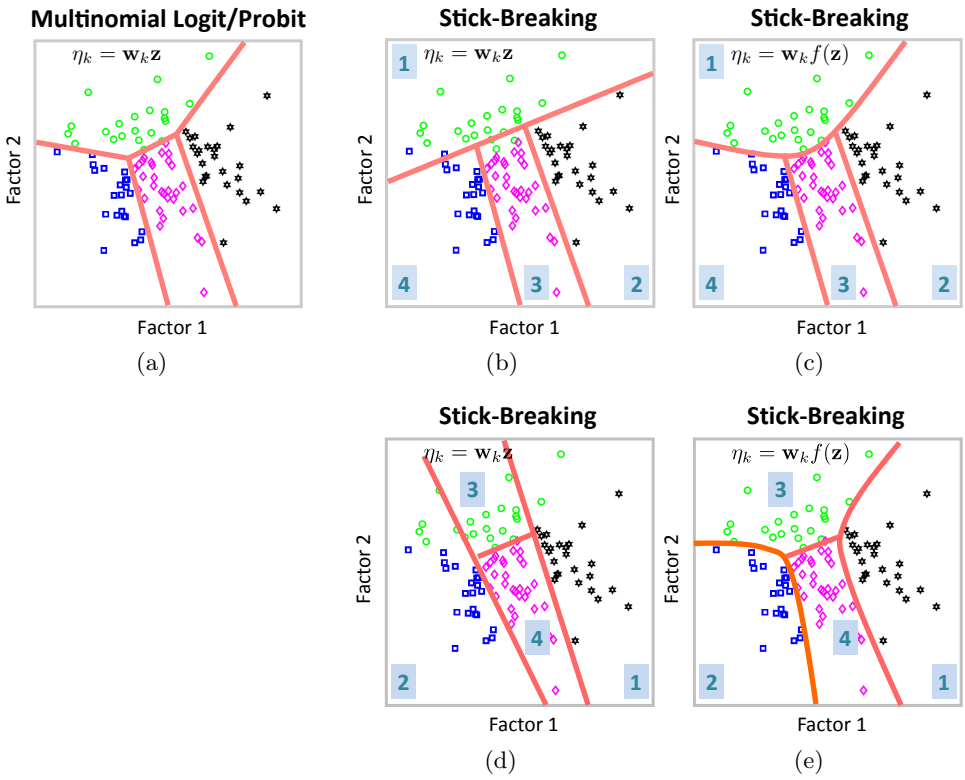


Figure 5.2: Illustrations showing decision boundaries. There are two features and 4 categories. In each figure, each point is a data example and its color (and marker) shows its category. The decision boundaries are shown with orange lines. The ordering of categories for stick breaking likelihood is indicated in blue boxes with numbers. Fig. (a) shows boundaries obtained with the multinomial logit/probit likelihood. Note that there is no ordering constraints imposed on the categories. Fig. (b) and (c) show boundaries for the stick breaking likelihood given a particular ordering of categories. Fig. (b) shows linear decision boundaries, while Fig. (c) shows non-linear boundaries. Fig. (d) and (e) shows the same for a different ordering of categories. This illustration shows that the stick breaking likelihood with linear features is unable to separate the data as well as the multinomial likelihood, however a good separation can be obtained with non-linear features (quadratic in this illustration).

this function with more than two categories, and all the bounds discussed before can be inaccurate. As we can see in Eq. 5.35, the stick-breaking

parameterization only depends on functions of the LLP functions $\log(1+e^{\eta_j})$. In stark contrast to the multinomial-logit case, accurate piecewise-linear and quadratic bounds are available for the LLP function; see Section 4.5. In addition, the quadratic bounds discussed in Chapter 4 can also be applied to the stick-breaking likelihood to obtain less accurate but faster algorithms. For example, the Bohning bound when applied will lead to a fast variational learning algorithm. We consider only piecewise bounds since our goal is to demonstrate that more accurate variational lower bounds lead to better learning algorithms.

5.6.1 Variational Learning Using Piecewise Bounds

The expectation of the log-likelihood for stick breaking parameterization is shown in Eq. 5.36. Here, $q(\eta|\widetilde{\gamma}_k) = \mathcal{N}(\eta|\widetilde{m}_k, \widetilde{v}_k)$ is the marginal Gaussian distribution of k'th predictor. The intractability arises due to the LLP function, and we substitute piecewise linear/quadratic bounds, described in Section 4.5, to get a tractable lower bound shown in Eq. 5.37.

$$\mathbb{E}_{q(\boldsymbol{\eta}|\widetilde{\boldsymbol{\gamma}})}[\log p(y|\boldsymbol{\eta})] = \mathbf{y}^T\widetilde{\mathbf{m}} - \sum_{j\leq k}\mathbb{E}_{q(\eta|\widetilde{\gamma}_k)}[\log(1+e^{\eta})] \tag{5.36}$$

$$\geq \mathbf{y}^T\widetilde{\mathbf{m}} - \sum_{j\leq k}\sum_{r=1}^{R}\overline{f}_r(\widetilde{m}_k, \widetilde{v}_k, \boldsymbol{\alpha}) \tag{5.37}$$

As discussed before, an important property of the piecewise bound is that its maximum error is bounded and can be driven to zero by increasing the number of pieces. This means that the evidence lower bound can be made arbitrarily tight by increasing the number of pieces, leading to an accurate estimation.

Similar to the binary case, we can optimize the lower bound with a gradient based approach. Computation of the gradients $\mathbf{g}^m_{dn}$ and $\mathbf{G}^v_{dn}$ is $O(DKNR)$. Compute the sum over d in the E and M-steps costs $O(NDKL^2)$. Inversion in E-step costs $O(NL^3)$. The total computational complexity of one EM step therefore is $O(DKNR + (DKL^2 + L^3)N)$.

5.7 Results

In this section, we compare performance on many datasets and models. We first compare the performance of the variational learning on synthetic data,

demonstrating that a more accurate LVB leads to a better variational algorithm. We then compare performance of variational learning to other existing methods, such as MCMC and variational Bayes, on real world datasets.

Synthetic data experiments

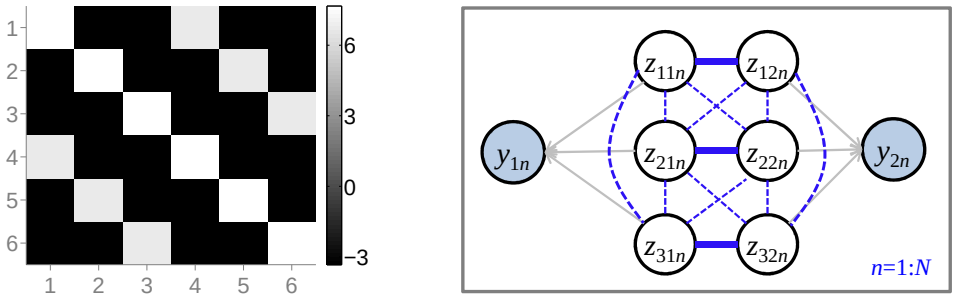
In this section, we compare performances of variational methods in modeling discrete distributions. Throughout this section, we use $p_{logit}(\mathbf{y}|\boldsymbol{\theta})$ to refer to the exact probability of a data vector $\mathbf{y}$ under the multinomial logit LGM with parameters $\boldsymbol{\theta}$. Similarly, we use $p_{stick}(\mathbf{y}|\boldsymbol{\theta})$ to refer to the exact probability under the stick breaking LGM. These exact probabilities remain intractable, but for small dimensions we can compute them to a reasonable accuracy using a Monte Carlo approximation to the integral, $p(\mathbf{y}|\boldsymbol{\theta}) = \int p(\mathbf{y}|\mathbf{z},\boldsymbol{\theta})\mathcal{N}(\mathbf{z}|\boldsymbol{\mu},\boldsymbol{\Sigma})d\mathbf{z}$. Our goal is to obtain estimates $\hat{\boldsymbol{\theta}}$ and compare $p_{logit}(\mathbf{y}|\hat{\boldsymbol{\theta}})$ and $p_{stick}(\mathbf{y}|\hat{\boldsymbol{\theta}})$ to the true discrete distribution.

We generate data from a 2D categorical latent Gaussian graphical model (cLGGM). We assume that both dimensions have $K+1$ categories. We set the predictor for the first category to 0, and use K latent variables to predict rest of the K categories. Define, $\mathbf{z}_n = (z_{11n}, z_{21n}, \ldots, z_{K1n}, z_{12n}, z_{22n}, \ldots, z_{K2n})$ as the vector containing latent variables for both dimensions. Since this is an LGGM model, we set $\mathbf{W}$ to identity and $\mathbf{w}_0$ to $\mathbf{0}$. We set the true mean parameter $\boldsymbol{\mu}^* = \mathbf{0}$. We define the true covariance parameter $\boldsymbol{\Sigma}^*$ for $\mathbf{z}_n$ as follows,

$$\Sigma^*_{ij} = \begin{cases} 20\,\mathrm{var}(X_{:,i}) + 1 & \text{if } i = j \\ 20\,\mathrm{covar}(X_{:,i}, X_{:,j}) & \text{if } i \neq j \end{cases} \tag{5.38}$$

where $\mathbf{X} = [I_K I_K]$, $\mathrm{var}(\mathbf{x})$ is the variance of vector $\mathbf{x}$, $\mathrm{covar}(\mathbf{x},\mathbf{y})$ is the covariance between $\mathbf{x}$ and $\mathbf{y}$. This choice of $\boldsymbol{\Sigma}_*$ forces both dimensions to take the same category, resulting in high correlation between dimensions. An example of $\boldsymbol{\Sigma}_*$ for $K = 3$ is shown in Fig. 5.3(a). Fig. 5.3(b) shows the graphical model for this cLGGM, where we show positive correlations between the latent variables with solid lines and negative correlations with dashed lines.

We sample 10^6 data cases from the multinomial logit model, and estimate parameters $\hat{\boldsymbol{\theta}}$ for both the multinomial logit and stick breaking LGMs. For the multinomial logit model, we use two versions of variational EM algorithms based on the Bohning bound and the log bound, respectively. For the stick model, we use our proposed variational EM algorithm. We refer to these three methods as 'logit-Bohning', 'logit-Log', and 'stick-PW' respectively. Note that since the data is generated from a multinomial logit



(a) (b)

Figure 5.3: A 2D categorical LGGM with $K = 3$. Fig. (a) shows the covariance matrix of the latent variables. Note that first 3 latent variables are for the first dimension, and the last 3 are for the second dimension. Fig. (b) shows the graphical model for the model. We show positive correlations between the latent variables with solid lines and negative correlations with dashed lines.

model, there is a modeling error for the stick breaking model in addition to the approximation error in learning. We will see below that, despite this error, the stick breaking likelihood models the data much better than other methods.

We first compare results for $K = 3$ in Fig. 5.4 which shows the true $p_{logit}(\mathbf{y}|\boldsymbol{\theta}_*)$ as well as $p_{logit}(\mathbf{y}|\hat{\boldsymbol{\theta}})$ for logit-Log and logit-Bohning, and $p_{stick}(\mathbf{y}|\hat{\boldsymbol{\theta}})$ for stick-PW. Note that the number of possible discrete data-vector is $2^{K+1} = 16$ and the figure shows the probabilities of these 16 data vectors. We see that stick-PW obtains a very close probability distribution to the true distribution, while other methods do not. Figure 5.5 shows results for $4, 5, 6, 7, 8$ categories. Here we plot KL-divergence between the true distribution and the estimated distributions for each method. We see that our proposed method consistently gives very low KL divergence values (the values for other methods are decreasing because the entropy of the true distribution decreases since we have set the multiplying constant in $\boldsymbol{\Sigma}_*$ to 20 for all categories).

Multi-class Gaussian process classification

In this section, our goal is to compare the marginal likelihood approximation and its suitability for parameter estimation. We consider a multi-class Gaussian process classification (mGPC) model. Since there are only 2 hyper-

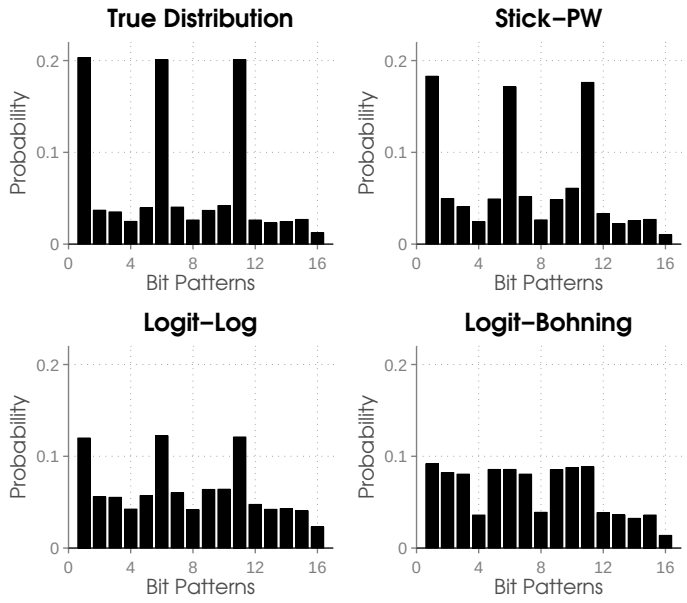


Figure 5.4: Comparison of the true probability distribution to the estimated distributions on synthetic data with 4 categories

parameters, we can compare marginal likelihood approximations for many hyper-parameter settings.

To get the 'true' value of the marginal likelihood, we use hybrid Monte Carlo (HMC) sampling along with annealed importance sampling (AIS). We apply this to the multinomial logit likelihood and refer to the truth as 'logit-HMC'. We compare to the multinomial probit model of Girolami and Rogers [2006], which uses variational-Bayesian inference. For this method, we use the MATLAB code provided by the authors. We refer to this as the 'probit-VB' approach. We also compare to multinomial-logit models learned using a variational EM algorithm based on the log bound and the Bohning bound. We refer to these methods as the 'logit-Log' and 'logit-Bohning' respectively. Finally, we call our proposed method with stick breaking likelihood as 'stick-PW'.

We apply the mGPC model to the forensic glass data set (available from the UCI repository) which has $D = 214$ data examples, $K = 6$ categories, and features of length 8. We use 80% of the dataset for training and the rest for testing. We set $\boldsymbol{\mu} = 0$ and use a squared-exponential kernel, for which the (i, j)'th entry of $\boldsymbol{\Sigma}$ is defined as: $\boldsymbol{\Sigma}_{ij} = -\sigma^2 \exp[-\frac{1}{2}||\mathbf{x}_i - \mathbf{x}_j||^2/s]$. To compare the marginal likelihood, we fix $\boldsymbol{\theta}$ which consists of

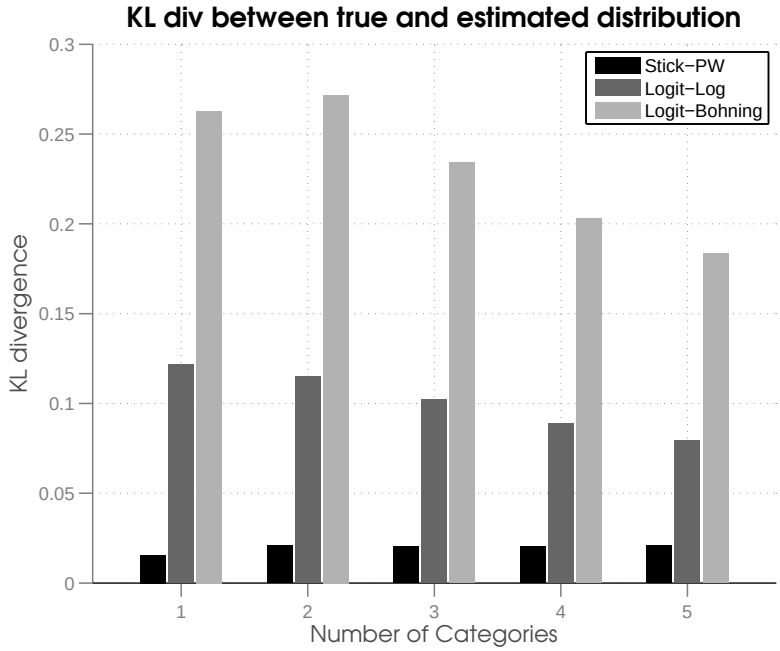


Figure 5.5: KL divergence between the true and estimated distributions for different categories.

σ and s and compute an approximation to the marginal likelihood using methods mentioned earlier. We compute the prediction error defined as $-\log_2 \tilde{p}(y_{test}|\boldsymbol{\theta}, \mathbf{y}_{train}, \mathbf{x}_{train}, \mathbf{x}_{test})$, where $(\mathbf{y}_{train}, \mathbf{x}_{train})$ and $(y_{test}, \mathbf{x}_{test})$ are training and testing data, respectively. Here, $\tilde{p}(y_{test}|\cdot)$ is the marginal predictive distribution approximated using the Monte Carlo method.

Figure 5.6 shows the contour plots for all the methods over a range of settings of hyperparameters. In each figure, top plot shows the negative log marginal likelihood approximation and the bottom plot shows the prediction error. The star indicates the hyperparameter value at the minimum of the negative log marginal likelihood. The Figure 5.6(a) shows the 'true' marginal likelihood obtained by logit-HMC. This plot shows the expected behavior of the true marginal likelihood. As we increase σ, we move from Gaussian-like posteriors to a posterior that is highly non-Gaussian. The posterior in the high σ region is effectively independent of σ and thus we see contours of marginal likelihood that remain constant (this has also been noted by Nickisch and Rasmussen [2008]). Importantly for model selection, there is a correspondence between the minimum value of the marginal

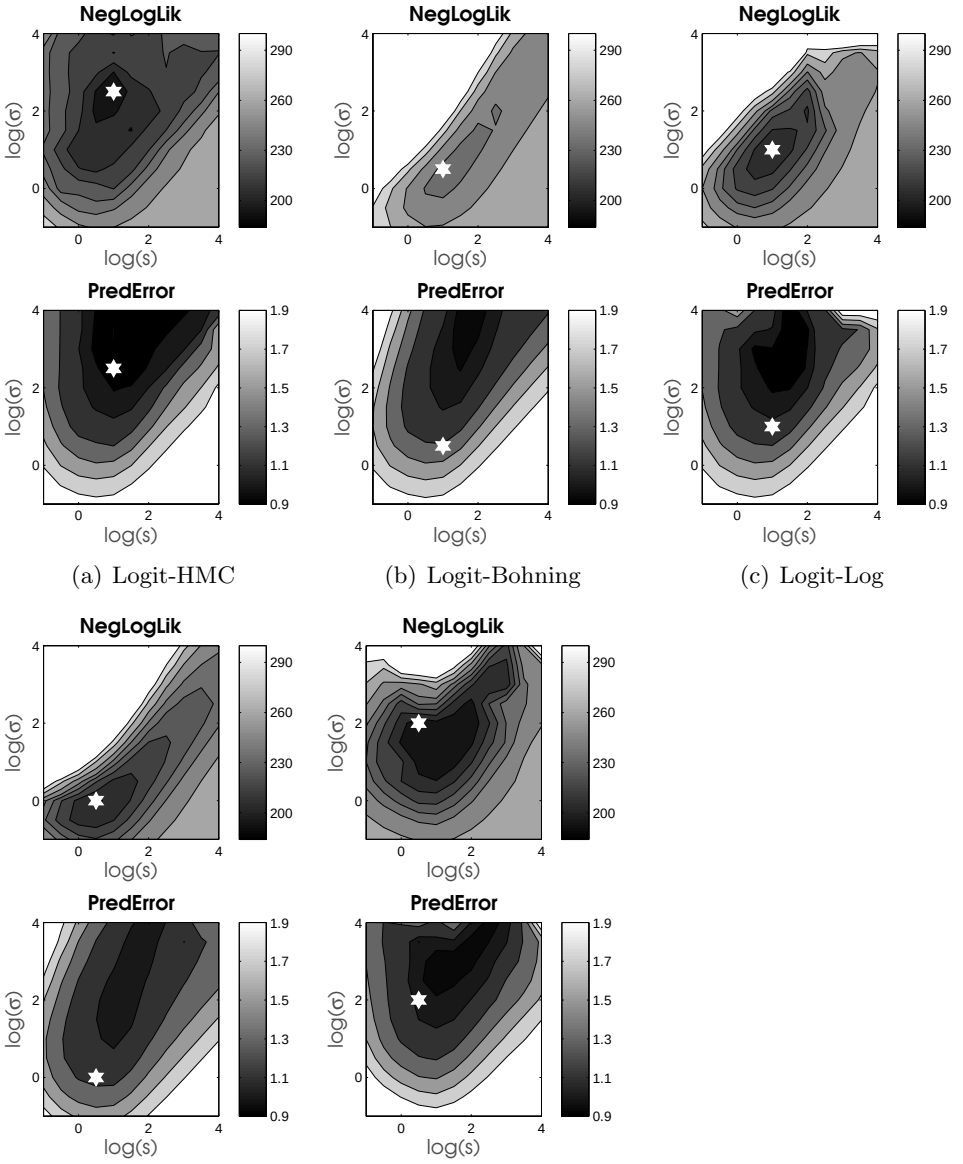
likelihood (or evidence) and the region of minimum prediction error. Thus optimizing the hyperparameters and performing model selection by minimizing the marginal likelihood gives optimal prediction. In our experience, tuning HMC parameters is a tedious task for this model as these parameters depend on $\boldsymbol{\theta}$. In addition, convergence is difficult to assess. Both HMC and AIS samplers need to be run for many iterations to get reasonable estimates.

Figure 5.6(b) and 5.6(c) show the negative of the log-marginal likelihood, along with the predictions for logit-Bohning and logit-Log, respectively. As we increase σ, the posterior becomes highly non-Gaussian and the variational bounds strongly underestimate the marginal likelihood in these regions (upper left corner of plots). The variational approximation also reduces the correspondence between the marginal likelihood and the test prediction, thus the minimum of the negative of the log-marginal likelihood is not useful in finding regions of low prediction error, resulting in suboptimal performance. The log bound, being a tighter bound than the Bohning bound, provides improved marginal likelihood estimates as expected, and a better correspondence between the prediction error and the marginal likelihood. Figure 5.6(d) is the behavior of the multinomial probit model and confirms the behavioral similarity of the logit and probit likelihoods.

The behavior of the stick likelihood is shown in Figure 5.6(e). The piecewise bound is highly effective for this model and the model provides good estimates even in the highly non-Gaussian posterior regions. An important appeal of this model is that the correspondence between the marginal likelihood and the prediction is better maintained than the logit or probit models, and thus parameters obtained by optimizing the marginal likelihood will result in good predictive performance.

We also experimented with different category orderings for this data to analyze the bias introduced by the ordering constraint. Although there are some orderings that lead to lower error than what we report here, the magnitude of the variance of this error is quite low. Hence, the stick likelihood still performs better than other methods. In general, we recommend a little experimentation with different ordering to find out if the data is sensitive to the ordering constraint.

Performance of all methods at the best parameter setting is summarized in Table 5.1 showing the best parameter values, approximation to the negative marginal log-likelihood, and prediction error.



(a) Logit-HMC (b) Logit-Bohning (c) Logit-Log

(d) Probit-VB (e) Stick-PW

Figure 5.6: Comparison of methods for multi-class GP classification on the Glass dataset using (a) Multi-HMC (b) Multi-Log (c) Multi-Bohning (d) Probit-VB (e) Stick-PW. For each method, the top plot shows negative of the log marginal likelihood approximations and the bottom plot shows prediction errors. Multi-HMC can be considered as the ground truth, so each method is compared against Figure (a).

Method	s	σ	negLogLik	predError
Logit HMC	1	2.5	198.63	0.92
Logit-Boh	1	0.5	239.28	1.31
Logit-log	1	1	208.26	1.13
Probit-VB	0.5	0	203.59	1.23
Stick-PW	0.5	2	194.16	1.07

Table 5.1: Performance at best parameter setting (a star in Figure 5.6)

.

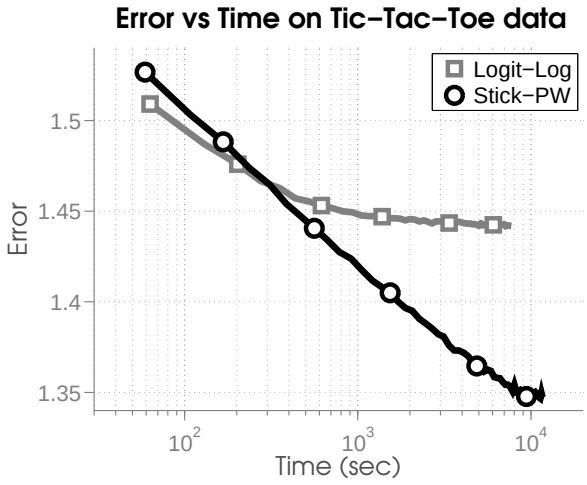


Figure 5.7: Imputation error vs time for cLGGM model on tic-tac-toe data.

Categorical latent Gaussian graphical model

We compare the logit-Log method to our proposed stick-PW method on the latent Gaussian graphical model. We use two datasets. The first dataset is the tic-tac-toe data set, which consists of 958 data examples with 10 dimensions each. All dimensions have 3 categories except the last one which is binary (thus the sum of categories used in the cLGGM is 29). The second data set is the ASES data set consists of survey data from respondents in different countries. We select the categorical responses for one country (UK), resulting in 17 response fields from 913 people; 9 response fields have 4 categories and the remainder have 3 categories. For both datasets, we use 80% of the data for training and rest for testing. We randomly introduce missing values in the test data, and compare cross entropy errors for these missing values.

Figure 5.7 shows the error versus time for one data split of the tic-tac-toe data. The plot shows that the stick-PW is a better method to use, since it gives much lower error when the two methods are run for the same amount of time. Figure 5.8(a) compares the error of the log bound and the piecewise bound for all 20 data splits used. For all splits, the points lie below the diagonal line, indicating that the piecewise bound has better performance. We show a similar plot for the ASES data set in figure 5.8(b), which more markedly shows the improvement in prediction when using the piecewise bound over the log bound.

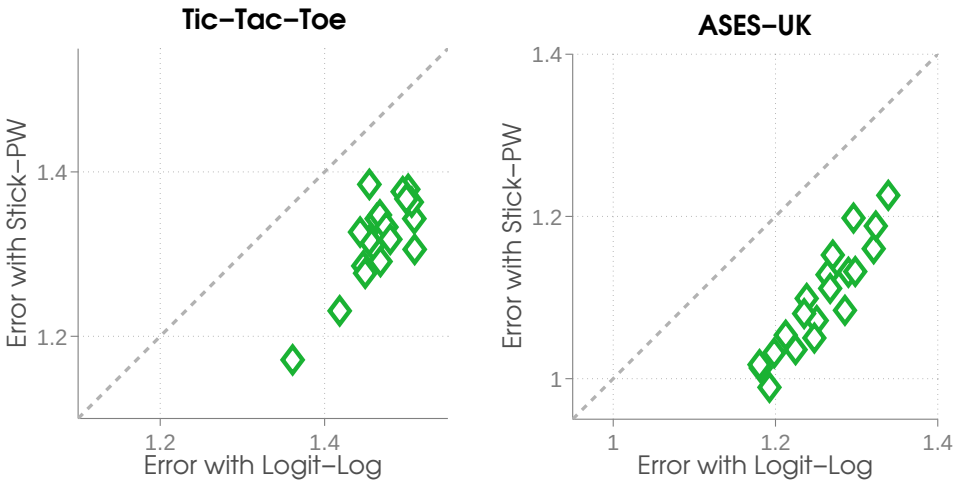


Figure 5.8: Imputation errors for Tic-tac-toe and ASES-UK datasets. Each point is a different train-test split and a point below the dashed line indicates that Stick-PW performs better than Multi-Log.

Chapter 6

Extensions and Future Work

In this chapter, we present extensions of our approach and discuss future work. We extend our approach to model ordinal data and to model data vectors containing mixed-type of variables. We briefly discuss modification of the variational algorithm when data vectors have missing entries. We discuss possible future directions to make the variational approach more generally applicable and computationally efficient.

6.1 Variational Learning for Ordinal Data

In Section 1.3.4, we discussed a variety of likelihoods for ordinal data. We now discuss application of our variational approach to some of those likelihoods.

We start with the cumulative logit model, which defines the cumulative probabilities of an observation y taking a value less than k as $\mathbb{P}(y \leq k|\eta) = \sigma(\phi_k - \eta)$, where η is the predictor, ϕ_k are real thresholds such $-\infty = \phi_0 < \phi_1 < \ldots < \phi_K = \infty$, σ is the sigmoid function $\sigma(x) := 1/(1 + \exp(x))$, and k is a category in the set $\{0, 1, \ldots, K\}$. Using this, the probability of $y = k$ can be expressed as shown below,

$$\mathbb{P}(y = k|\eta) = \sigma(\phi_k - \eta) - \sigma(\phi_{k-1} - \eta) \tag{6.1}$$

$$= \frac{1}{1 + e^{-(\phi_k - \eta)}} - \frac{1}{1 + e^{-(\phi_{k-1} - \eta)}} \tag{6.2}$$

$$= \frac{e^{\eta}\left(e^{-\phi_{k-1}} - e^{-\phi_k}\right)}{\left[1 + e^{-(\phi_k - \eta)}\right]\left[1 + e^{-(\phi_{k-1} - \eta)}\right]} \tag{6.3}$$

Taking log, the log-likelihood can be written in terms of the LLP function,

$$\log p(y = k|\eta) = \eta - \mathrm{llp}(\eta - \phi_k) - \mathrm{llp}(\eta - \phi_{k-1}) - \log(e^{\phi_{k-1}} - e^{\phi_k}) \tag{6.4}$$

Recall that $\mathrm{llp}(x) = \log(1 + \exp(x))$.

For variational learning, we need to be able to compute a lower bound to expectation of the log-likelihood with respect to the Gaussian distribution

$q(\eta|\widetilde{\boldsymbol{\gamma}}) = \mathcal{N}(\eta|\widetilde{m}, \widetilde{v})$. For cumulative logit likelihood, this term takes the following form,

$$\mathbb{E}_{q(\eta|\widetilde{\boldsymbol{\gamma}})}[\log p(y=k|\eta)] = \\ \widetilde{m} - \mathbb{E}_{q(\eta|\widetilde{\boldsymbol{\gamma}})}[\mathrm{llp}(\eta-\phi_k)] - \mathbb{E}_{q(\eta|\widetilde{\boldsymbol{\gamma}})}[\mathrm{llp}(\eta-\phi_{k-1})] - \log(e^{\phi_{k-1}} - e^{\phi_k}) \quad (6.5)$$

We can see that this term is intractable due to the expectation of the LLP function in second and third terms, respectively. For a tractable lower bound, we can use the piecewise linear/quadratic upper bounds to the LLP function. The parameters $\Phi = \{\phi_1, \phi_2, \ldots, \phi_{K-1}\}$ can be estimated easily by taking derivatives of this lower bound. The ordering constraint over Φ can be handled by reparameterizing them as $\phi_k - \phi_1 = \sum_{l=2}^{k} \exp(t_l), 1 < k < K$ with $t_k \in \mathbb{R}$, and estimating t_k instead.

Next, we consider the continuation ratio model. One way to define a continuation ratio model is to define probabilities as $p(y=k|\eta) := 1/(1+\exp(\phi_k - \eta))$. We can check that, under this definition, the ratio $\mathbb{P}(y=k|\eta)/\mathbb{P}(y>k|\eta)$ is equal to $\phi_k - \eta$, making it a continuation ratio model. The log-likelihood is simply the negative of the LLP function, $\log p(y=k|\eta) = -\log(1+\exp(\phi_k - \eta))$. Here again, the tractable variational lower bound can be done using the piecewise bounds.

Finally, consider the stereotype regression model. The likelihood and its log are shown below,

$$p(y=k|\eta) = \frac{\exp(\phi_k - \alpha_k \eta)}{\sum_{j=0}^{K} \exp(\phi_j - \alpha_j \eta)} \quad (6.6)$$

$$\log p(y=k|\eta) = \phi_k - \alpha_k \eta - \mathrm{lse}(\boldsymbol{\phi} - \eta\boldsymbol{\alpha}) \quad (6.7)$$

where $\boldsymbol{\phi}$ and $\boldsymbol{\alpha}$ are the vectors of ϕ_k and α_k respectively. For tractable variational learning, we can use upper bounds on the LSE function discussed in Chapter 5.

6.2 Variational Learning for Mixed Data

Many real worlds datasets contain observations of mixed types. For example, survey datasets in social science contain continuous (e.g. respondent's weight, age and salary), binary (e.g. respondent's gender), categorical (e.g. respondent's country or city), and ordinal (e.g. how strongly does a respondent agrees with a government policy). An analysis of such dataset involves learning correlations between these variables. For example, we might be interested in assessing the effect of age and salary of a respondent to his/her

views on government policy. A factor analysis (FA) model is usually used to jointly learn such correlations Khan et al. [2010].

Our proposed variational framework can easily handle the datasets containing mixed type of variables. The ELBO of Eq. 3.8 in Section 3.1 essentially remains the same for the mixed data vectors, except that the summation over d now consists of mixed type of log-likelihoods. Hence, for a tractable lower bound, we employ many types of LVBs, instead of using only one type as before. See Khan et al. [2010] for a detailed derivation.

We now present an example of mixed data FA. Our goal in this example is to illustrate the use of mixed-data FA. We compare with two other models which avoid a proper modeling of mixed data. First approach is to fit Gaussian FA using only continuous variables and ignoring all the discrete variables. We call this approach 'GaussFA-1'. Second approach is to fit Gaussian FA, but now also using the discrete variables recoded as continuous variables. For recoding, we first encode a discrete variable in dummy encoding, e.g. a 3 category variable is recoded as $(1, 0, 0)$ for category 1, $(0, 1, 0)$ for category 2, and $(0, 0, 1)$ for category 3. Then, we transform each binary element of the dummy encoding to $\{-1, 1\}$, e.g. $(1, 0, 0)$ becomes $(1, -1, -1)$. We call this approach 'GaussFA-2'. Note that there is no learning error in these models, since we can use the exact EM algorithm. For our mixed-data FA model, we use Gaussian likelihoods for continuous variables, and multinomial logit likelihoods for discrete variables. We fit the model using the variational EM algorithm based on the Bohning bound.

We fit FA models to the Auto dataset which contains data about 392 cars. Each data vector consists of 5 continuous and 3 discrete variables with 3,5, and 13 categories, respectively. The 5 continuous variables are the following: 'miles-per-gallon', 'displacement', 'horsepower', 'weight', and 'acceleration' The 3 discrete variables are the following: 'number-of-cylinders', 'model-year', and 'country-of-origin'.

For all the three FA models, we use only two latent factors. Figure 6.2 shows the results. We plot the posterior means of latent factors for all the data examples; each data example here is a car. The posterior variance is quite small (in the range of 0.05-0.8), so we do not include it in the plot. To make the interpretation easy, we color code each car depending on the country it belongs to ('country-of-origin' is one of the discrete variable).

We note that each method shows some clustering between the American cars and the non-American ones. Even GaussFA-1 shows some separation, which means that the continuous variables are informative about the clustering. However, inclusion of additional recoded discrete variables in GaussFA-2 does not improve the clustering. MixedDataFA, being a prin-

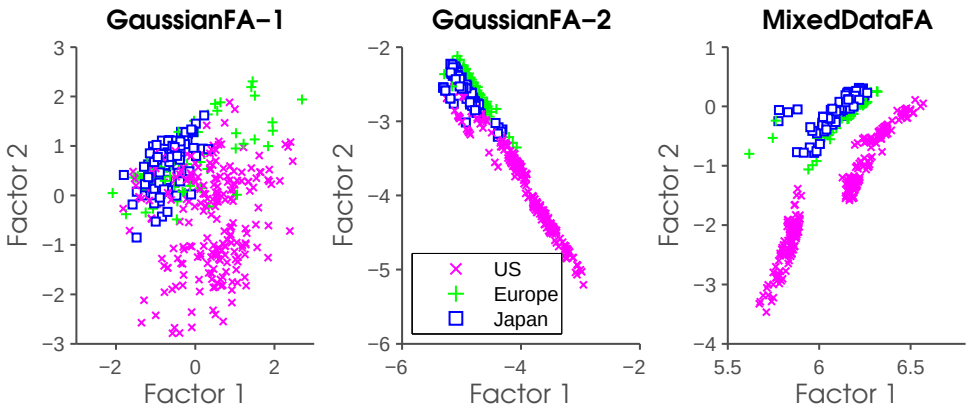


Figure 6.1: Visualization of a 2 factor FA model learned using the Auto data. We plot the posterior means of latent factors for all cars. For easy interpretation, we color code each car depending on its country of origin.

cipled approach for combining the discrete and continuous variables, shows very "clean" clustering between the American and non-American cars.

6.3 Variational Learning with Missing Data

Our variational approach can be easily modified to handle missing entries. Denote the set of observed dimensions in n'th data vector by O_n, and the set of data vectors with d'th dim observed by O_d. The lower bound of Eq. 3.21 is modified to contain only the observed dimensions in summation over d, as shown below,

$$\underline{\mathcal{L}}_n(\boldsymbol{\theta}, \boldsymbol{\gamma}_n, \boldsymbol{\alpha}_n) := \tfrac{1}{2}\left[\log|\mathbf{V}_n\boldsymbol{\Sigma}^{-1}| - \mathrm{tr}(\mathbf{V}_n\boldsymbol{\Sigma}^{-1}) - (\mathbf{m}_n - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{m}_n - \boldsymbol{\mu}) + L\right] + \sum_{d \in O_n} \underline{f}(y_{dn}, \widetilde{\gamma}_{dn}, \boldsymbol{\alpha}_{dn}) \tag{6.8}$$

The gradients in Algorithm 1 are also modified accordingly. All summations over d are changed to $d \in O_n$ and all summations over n are changed to $n \in O_d$. Using this, the modification of variational algorithms for particular bounds is straightforward.

6.4 Future Work

As we discussed earlier, there are two major challenges with the variational learning in LGMs. First, the intractability of ELBO, and second, its computational inefficiency. In this thesis, we discussed solutions for both aspects. We now discuss possible future directions for making variational learning more generally applicable and computationally efficient.

6.4.1 Designing Generic LVBs

The accuracy of variational learning depends heavily on the accuracy of the local variational bounds to $\mathbb{E}_{q(\eta|\widetilde{\gamma})}[\log p(y|\eta)]$. In this thesis, we derived accurate LVBs for several likelihoods. It is not clear how to generalize such design procedures for general likelihoods. Design of clever quadrature techniques, where we compute expectations by exploring high density regions of the Gaussian distribution, seems plausible. However, this might be limited by the dimensionality of the Gaussian distribution. A recent approach by Chai [2012] for multinomial logit likelihood is promising, and a generalization of this approach will be extremely useful. Another recent effort is made by Paisley et al. [2012], where stochastic approximation to the expectation term is made. The usual problem in such approaches is the difficulty in assessment of the approximation error. If an estimate of the error is available, it will help us design strategies to increase the accuracy as the algorithm progresses, for example, by increasing the number of samples. This problem is similar to the one discussed in Section 2.2 for stochastic EM algorithms. Therefore, it is important to realize that not only we require a good approximation, we must also be able to asses or bound the error made in the approximation. We showed in this thesis that it is possible to have such error guarantees sometimes. A generalization of our approach, although difficult, will definitely be useful.

6.4.2 Generalization to Other Likelihoods

There are many other interesting likelihoods for which the $\mathbb{E}_{q(\eta|\widetilde{\gamma})}[\log p(y|\eta)]$ term is not available in closed form, e.g. the mixture of multinomials in correlated topic model. Extensions to these likelihoods will be extremely useful.

Another issue is the concavity of $\mathbb{E}_{q(\eta|\widetilde{\gamma})}[\log p(y|\eta)]$ with respect $(\widetilde{m}, \widetilde{v})$. Exact conditions under which such concavity holds are not known. Recently, Challis and Barber [2011] proved that the expectation term is concave with

respect to $(\widetilde{m}, \sqrt{\widetilde{v}})$, if $\log p(y|\eta)$ is concave with respect to η. Extending our methods to work with concavity with respect to $\sqrt{\widetilde{v}}$ will generalize our results to log-concave likleihoods.

Note that concavity of the expectation term, although useful, is not always necessary to design efficient algorithms. In fact, if this term is unimodal with respect to $(\widetilde{m}, \widetilde{v})$ and if we can find a concave lower bound which is tight at one point, we can still design efficient concave algorithms using a methodology similar to the majorization-minorization algorithms [Hunter and Lange, 2004]. Future work in this direction will help generalize the variational approach.

6.4.3 Approximate Gradient Methods

We showed in Section 3.4 that the variational inference involves optimization of a concave function. This allows us to use the theory of approximate gradient algorithms to reduce the computation. It is well-known that approximate gradient methods can lead to linear convergence rates in case of concave functions; see Friedlander and Schmidt [2011] for example. These methods employ an approximation of the gradient, which is cheaper to compute. To be specific, given a concave function $f(\mathbf{x})$, we compute an approximation $\mathbf{g}_k := \bigtriangledown f(\mathbf{x}_k) + \mathbf{e}_k$ in the k'th iteration. A line search algorithm is guaranteed to convergence at a "good" rate using a sequence B_k which bounds the norm of the error $||\mathbf{e}_k||^2 \leq B_k$ for every iteration k. See Friedlander and Schmidt [2011] on details on designing B_k.

Below, we discuss a computationally cheap method of building gradient approximations and determining the magnitude of the error. We consider the inference algorithm discussed in Section 3.6 for the special LGMs such as GP and LGGM. Recall, from Eq. 3.48, that the gradient of ELBO, denoted here by $\mathbf{G}^v$, is given as follows,

$$\mathbf{G}^v = \tfrac{1}{2}\left(\mathbf{V}^{-1} - \mathbf{\Sigma}^{-1}\right) + \mathrm{diag}(\mathbf{g}^v) \tag{6.9}$$

The main computation bottleneck is the computation of $\mathbf{V}^{-1}$, which is $O(D^3)$.

We propose to compute an approximation of the gradient by replacing $\mathbf{V}^{-1}$ with an estimate of it. We do so by subsampling the data, i.e. we only consider a subset $S \subseteq \{1, 2, \ldots, D\}$ of data examples and define a new ELBO, which is a variant of the ELBO of Eq. 3.46 (we only keep the terms involving $\mathbf{V}$ for presentation clarity).

$$\tfrac{1}{2}\left[\log|\mathbf{V}\mathbf{\Sigma}^{-1}| - \mathrm{tr}(\mathbf{V}\mathbf{\Sigma}^{-1})\right] + \sum_{d\in S} \underline{f}(y_d, \gamma_d, \boldsymbol{\alpha}_d) \tag{6.10}$$

We see that the only difference between the original ELBO and the new one is in the summation over d. The new ELBO includes only the observations in S. Our plan is to optimize the above ELBO to get an estimate of the inverse.

First, we show that optimizing the new ELBO is much cheaper than optimizing the original ELBO. Let us consider a simple case where we subsample only first two observations, i.e. $S = \{1, 2\}$. The gradient of the new objective function can be written as following.

$$\mathbf{G}_S^v = \tfrac{1}{2}\left(\mathbf{V}^{-1} - \mathbf{\Sigma}^{-1}\right) + \begin{pmatrix} g_1^v & 0 & 0 & \dots & 0 \\ 0 & g_2^v & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix} \tag{6.11}$$

A great insight is that, at the solution, only the first two diagonal elements of $\mathbf{V}^{-1}$ need to be computed since other elements are equal to $\mathbf{\Sigma}^{-1}$. Therefore, the solution can be obtained using our fast inference algorithm, the complexity of which will be $O(|S|^3)$ where $|S|$ is cardinality of S.

Next, we would like to estimate the error in our gradient approximation. This is easy. We take the difference between $\mathbf{G}^v$ and $\mathbf{G}_S^v$ and take the trace norm to find the following expression for the error.

$$\mathrm{Tr}(\mathbf{G}^v - \mathbf{G}_S^v) = \sum_{d \notin S} g_i^v \tag{6.12}$$

The error depends on all the data examples that we left out. We can easily build an estimate of the error, for example, using bootstrap. This is efficient since g_i^v can be computed cheaply. Given a sequence of B_k's, we can easily design an approximate gradient method such that the error is bounded by B_k's at all k.

6.4.4 Large-scale Matrix Factorization

In the previous section, we introduced sparsity in the data by subsampling. We showed that sparse data allows us to design fast inference algorithms. There are practical applications where the data is inherently sparse. Matrix factorization for movie recommendation is one such application [Guo and Schuurmans, 2008; Salakhutdinov and Mnih, 2008a]. In movie recommendation, given movie ratings from different users, our goal is to predict ratings for new movies. Here, data is sparse since not all users rate all the movies.

LGMs, such as factor analysis and PCA, have been applied to this problem and have been shown to perform well. A proper Bayesian analysis usually is difficult since it is computationally intensive for large-scale data available for movie prediction. For example, Netflix dataset has more than 400,000 user ratings of more than 17,000 movies, leading to a slow Bayesian learning [Salakhutdinov and Mnih, 2008a]. An additional issue is that the discrete ratings are usually recoded as continuous variables and a Gaussian likelihood is used. This is necessary since the factor models with discrete likelihood do not scale well to large data. For data with N observations, D-dimensional features and L latent factors, the computational cost is $O(NL^3 + NDL^2)$ and the memory cost is $O(NL^2)$.

In this section, we present an algorithm for discrete-data matrix factorization with computation cost of $2D^2+O(\sum_n D_n^3)$ per iteration, where D_n is the number of observed rating for n'th user. We saw in the previous section that, for sparse data, inference can be carried out efficiently for LGGMs. We now present an efficient EM algorithm for parameter learning. Our preliminary analysis reveals that LGGM can potentially scale much better than its low dimensional counterparts such as PCA.

Let us first define some notation. We denote the number of observed variables in n'th data vector by D_n, and assume that $D_n << D$. Let $\mathbf{a}_O$ refer to the part of $\mathbf{a}$ that is indexed by set O, $\mathbf{\Sigma}_{OO}$ refer to the subblock of $\mathbf{\Sigma}$ formed by taking rows and columns indexed by O, $\mathbf{\Sigma}_{:O}$ refer to the subblock of $\mathbf{\Sigma}$ formed by taking all rows but only columns indexed by O, Let $[\mathbf{a}_O]$ be a vector of length D such that entries in O are equal to $\mathbf{a}_O$ with rest of the elements are zero. Similarly, $[\mathbf{A}_{OO}]$ is a matrix of size $D \times D$ formed by zero padding the smaller matrix $\mathbf{A}_{OO}$.

For simplicity, we assume that $\boldsymbol{\mu} = 0$ and we need to estimate only $\mathbf{\Sigma}$. In the E-step, we compute the posterior mean $\mathbf{m}_n$ and covariance $\mathbf{V}_n$. Similar to previous section (see Eq. 6.11), fixed point updates of $\mathbf{m}_n$ and $\mathbf{V}_n$ can be rewritten as follows,

$$-\mathbf{\Sigma}^{-1}(\mathbf{m}_n - \boldsymbol{\mu}) + [\mathbf{g}_{nO}^m] = 0 \tag{6.13}$$

$$\tfrac{1}{2}\left(\mathbf{V}^{-1} - \mathbf{\Sigma}^{-1}\right) + \mathrm{diag}([\mathbf{g}_{nO}^v]) = 0 \tag{6.14}$$

Here, $\mathbf{g}_{nO}^m$ and $\mathbf{g}_{nO}^v$ are gradients of LVBs with respect to $\mathbf{m}_n$ and $\mathrm{diag}(\mathbf{V}_n)$ respectively, but only for observed dimensions. These are then zero padded to match the dimensions to the fully observed case. For these fixed points, the cost can be reduced to $O(D_n^3)$ as explained in the previous section.

For M-step, the update for $\mathbf{\Sigma}$ is given as shown below.

$$\mathbf{\Sigma} = \sum_{n=1}^{N} \mathbf{V}_n + \mathbf{m}_n \mathbf{m}_n^T \tag{6.15}$$

There are two issues with this update. First, we have to explicitly form the matrix $\mathbf{V}_n$ which is a $D \times D$ matrix. Second, computation of the product and sum is of the order $O(ND^2)$ which can be huge for large N and D. We now show that we can avoid these expensive steps and reduce the computation cost to $O(\sum_n D_n^2)$.

From the fixed-point equation of $\mathbf{V}_n$ in Eq. 6.14, we can see that the solution $\mathbf{V}_n$ will take the form shown below and then simplified further in terms of a smaller matrix $\mathbf{B}_n := \mathbf{\Sigma}_{OO} + \text{diag}(\mathbf{h}_{nO})$ of size $D_n \times D_n$.

$$\begin{aligned}
\mathbf{V}_n &= \left(\mathbf{\Sigma}^{-1} - 2\text{diag}([\mathbf{h}_{nO}])\right)^{-1} && (6.16)\\
&= \mathbf{\Sigma} - \mathbf{\Sigma}_{:O} \left(\mathbf{\Sigma}_{OO} - 2\text{diag}(\mathbf{h}_{nO})\right)^{-1} \mathbf{\Sigma}_{O:} && (6.17)\\
&= \mathbf{\Sigma} - \mathbf{\Sigma}[\mathbf{B}_n]^{-1}\mathbf{\Sigma} && (6.18)
\end{aligned}$$

A similar expression for $\mathbf{m}_n$ can also be obtained: $\mathbf{m}_n = \boldsymbol{\mu} + \mathbf{\Sigma}_{:O}\mathbf{g}_{nO}$. We substitute these in update of $\mathbf{\Sigma}$ in Eq. 6.15, to get the following simplification,

$$\mathbf{\Sigma} = \sum_{n=1}^{N} (\mathbf{V}_n + \mathbf{m}_n \mathbf{m}_n^T) = \mathbf{\Sigma} - \mathbf{\Sigma} \sum_{n=1}^{N} [\mathbf{B}_n^{-1}] \mathbf{\Sigma} \tag{6.19}$$

The advantage of this form is that we only have to compute $\mathbf{g}_{nO}$ and $\mathbf{B}_n^{-1}$ for each n and never have to form the matrix $\mathbf{V}_n$ completely. We also have to do only two multiplications of $D \times D$ once per EM step, instead of for every n. The complexity of M-step therefore reduces to $O(\sum_n D_n^2)$ additions, plus 2 D^2 multiplications. This is a huge reduction from $O(ND^2)$ multiplications. The total computation cost of the algorithm is $2D^2 + O(\sum_n D_n^3)$.

One problem with this approach might be the number of parameters in the models. In the factor analysis model, the number of parameters is $D \times L$ where L is the latent dimension and much smaller than D. The LGGM the number of parameters is D^2, much larger than the factor model.

Similar ideas have been explored by Yu et al. [2009] who consider the case of Gaussian LGGMs, and our approach generalizes their ideas to discrete-data LGMs, but in the context of variational learning. Yu et al. [2009] show a significant improvements in speed and accuracy over existing methods, but completely ignore the issue of large number of parameters. Perhaps it

is not an issue when N is much larger than D and there is enough data to be able to estimate D^2 number of parameters. Nevertheless, some overfitting is expected when N is not much bigger than D. This overfitting can be eliminated by forcing $\mathbf{\Sigma}$ to be sparse, for example, by assuming an l_1 prior. One has to search for a good regularization parameter, increasing the computational overhead. This is similar in spirit, though, to the search for number of factors in a factor model, which also increases computations. Which method performs better remains an open question.

6.4.5 Other Speed Ups

More speed-ups may be possible using following idea,

Parallel updates It is straightforward to obtain a parallel version of Algorithm 3 where we run fixed point updates for all dimensions parallelly, followed by one update of $\mathbf{V}$. Parallel updates are very popular for EP and have shown to improve the computational efficiency [Van Gerven et al., 2010], although neither the sequential or parallel version are provably convergent. Although the sequential version of our algorithm is provably convergent, it is not clear whether parallel updates will be convergent. A theoretical analysis might be possible since our objective function is concave. It is also required to know the conditions under which such an update will result in a positive definite matrix. Finally, we need to assess the computational gains is obtained in practice with such an update.

Sparse models Our EM algorithm can exploit sparsity in $\mathbf{\Omega}$ and can lead to further speed ups. There are many practical examples where $\mathbf{\Omega}$ is sparse; see Rue et al. [2009] for a few examples. In some applications, we would like to force $\mathbf{\Omega}$ to be sparse, perhaps because the underlying dependencies in the latent variables is sparse or because may be just to reduce number of parameters in the model. For such cases, we need better ways of being able to quantify the extent of sparsity required. For example, in case of l_1 prior for sparsity, how should we choose the strength of the prior? The marginal likelihood estimates might provide some clues in some scenarios, although its suitability need to be studied for a variety of applications.

Sparse posterior approximations Throughout this thesis, we assumed that the posterior covariances $\mathbf{V}$ are dense matrices. This assumption might be relaxed. In practice, instead of seeking a dense posterior, we might just want to keep only relatively high values of correlation. One possible way

to do achieve this, is to force a sparse prior on $\mathbf{V}$. This is a much less restrictive assumptions than the factorization assumption made in the mean field approximation, since here the factorization is learned from the data. The evidence lower bound discussed in this thesis can be easily modified to contain a sparse penalty term for $\mathbf{V}$. Given that the penalty is concave, the lower bound remains concave leading to an efficient algorithm. The computation complexity will depend on the strength of the prior, which needs to be estimated similar to the sparse model case. Such sparse approximations have recently been used in Yan and Qi [2010].

Online EM It is simple to extend the batch variational EM algorithms discussed in this thesis to the online setting, to handle the case of large N. See Cappe and Mouline [2009] and Hoffman et al. [2010].

Chapter 7

Conclusions

Modeling of high-dimensional, correlated, multivariate discrete data is an important problem in machine learning and computational statistics. In this thesis, we focused on the Bayesian modeling of such discrete data using LGMs. Our solutions were based on a variational approach which uses the evidence lower bound optimization. We made several contributions to the variational approach, focusing on the following two major aspects: tractability and computational efficiency. For tractability, we derived, applied, and compared many LVBs for tractable variational learning. Our work in this thesis clearly showed that accurate local bounds lead to a dramatic improvement in the accuracy of the variational approach. To improve computational efficiency, we used concavity of the variational lower bound, resulting in algorithms with a wide range of speed-accuracy trade-offs. Application to real-world datasets confirmed the significance of our contributions.

We started, in Chapter 1 with a generic definition of LGM which includes many popular models. We identified our learning objectives, dedicating the rest of the thesis to achieve them. The difficulty arises due to an intractable integral arising from the marginalization of latent variables. We reviewed several approaches in Chapter 2, which can be classified in three major categories: non-Bayesian methods, sampling methods, and deterministic methods. All of these approaches have their own advantages and disadvantages. Non-Bayesian approaches are fast, but they overfit. MCMC methods perform well, but are slow. Deterministic methods are faster than MCMC, but are less general than them. The variational approach falls in this last category and, as we showed in Chapter 3, is intractable for many discrete-data likelihoods. An additional issue is that, although it is faster than MCMC, it still has a lot of room for computational improvements.

Our first conclusion is that concavity is extremely useful for computational efficiency of the variational approach. In this regard, we made three contributions in Chapter 3. First, we established conditions under which the variational lower bound is concave. Second, we derived generalized gradient expressions for lower bound optimization. Third, using the concavity, we derived a fast convergent algorithm for variational inference. Our algo-

rithm borrows ideas from well-known concave problems, such as covariance selection and non-linear least squares, and efficiently exploits sparsity in the data and model parameters. Not only that, but it is amenable to many other speed-ups using approximate gradient methods and parallelization.

Our second conclusion is regarding the tractable LVBs and their affect on the accuracy of the variational approach. In Chapter 4 and 5, we derived and discussed many LVBs for binary and categorical data. We discussed extensions to ordinal and mixed-data in Chapter 6. We found that the accuracy of variational approach depends heavily on the accuracy of LVBs. Existing LVBs can lead to poor performance because of their large approximation error. We showed in this thesis, through the use of piecewise linear and quadratic bounds, that accurate bounds lead to huge improvements in accuracy. Finally, our error analysis revealed that an error assessment helps to choose appropriate LVBs for a given application.

Our final conclusion is that the variational approach can give rise to algorithms with a wide range of speed-accuracy trade-offs. The proposed Bohning bound leads to extremely fast, but sometimes inaccurate, variational algorithms. Our piecewise bounds, although slower than the Bohning bound, can trade-off speed for accuracy by increasing the number of pieces. In practice, this leads to efficient analysis. For example, for large datasets, quick results can be obtained with the Bohning bound and later can be refined by increasing the number of pieces in the piecewise bounds.

Although, we showed many positive results of the variational approach, one needs to be careful about its limitations. Just like other deterministic methods, our variational approach uses a Gaussian posterior approximation. This might lead to poor performance in cases where the posterior distribution is highly non-Gaussian. This is reflected in our experiments, most clearly seen in the results for Gaussian process classification; see Fig. 4.14 and 4.15 in Chapter 4. In that experiment, even though the variational lower bound underestimated the marginal likelihood sometimes, the prediction accuracy was still good. This experiment, and others like it, suggest that the situation may not be as bad as we might think. The key perhaps lies in the error in the Jensen bound and a careful study is required to understand this phenomenon.

In this thesis, we showed that our variational approach can be accurate and computationally efficient. We believe that this is just the tip of the iceberg. Deterministic methods, such us the one discussed in this thesis, are key to accurate and scalable learning. Today, there exist a variety of deterministic methods. They perform well at times and fail at others. However, in our experience, it is necessary to gain good understanding of strengths and

weaknesses of these deterministic methods, and combine them to design better methods. In the future, application of clever optimization techniques will help us design algorithms that are as fast as the non-Bayesian approaches. Design of accurate approximations and lower bounds will push deterministic methods to perform as well as MCMC algorithms. We hope that our work will motivate other researchers to work on making deterministic methods more efficient and widely applicable.

Bibliography

A. Agresti. *Analysis of ordinal categorical data*, volume 656. John Wiley & Sons Inc., 2010.

A. Ahmed and E. Xing. On tight approximate inference of the logistic-normal topic admixture model. In *International conference on Artificial Intelligence and Statistics*, 2007.

J. Ahn and J. Oh. A constrained EM algorithm for principal component analysis. *Neural Computation*, 15:57–65, 2003.

J. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422): 669–679, 1993.

C. Ananth and D. Kleinbaum. Regression models for ordinal responses: a review of methods and applications. *International journal of epidemiology*, 26(6):1323–1333, 1997.

J. Anderson. Regression and ordered categorical variables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 46(1):1–30, 1984.

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

D. J. Bartholomew. Factor analysis for categorical data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 42(3):pp. 293–321, 1980.

M. Beal. *Variational algorithms for approximate Bayesian inference*. PhD thesis, Gatsby Unit, 2003.

D. P. Bertsekas. *Nonlinear programming*. Athena Scientific, second edition, 1999.

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

D. Blei and J. Lafferty. Correlated topic models. In *Advances in Neural Information Processing Systems*, 2006.

D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

D. Bohning. Multinomial logistic regression algorithm. *Annals of the Institute of Statistical Mathematics*, 44:197–200, 1992.

J. Booth and J. Hobert. Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(1):265–285, 1999.

G. Bouchard. Efficient bounds for the softmax and applications to approximate inference in hybrid models. In *NIPS 2007 Workshop on Approximate Inference in Hybrid Models*, 2007.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge, 2004.

M. Braun and J. McAuliffe. Variational inference for large-scale models of discrete choice. *Journal of the American Statistical Association*, 105(489): 324–335, 2010.

D. Bunch. Estimability in the multinomial probit model. *Transportation Research Part B: Methodological*, 25(1):1–12, 1991.

W. Buntine. Variational extensions to EM and multinomial PCA. In *European Conference on Machine Learning*, pages 23–34. Springer, 2002.

O. Cappe and E. Mouline. Online EM algorithm for latent data models. *Journal of Royal Statistical Sociecty, Series B*, 71(3):593–613, June 2009.

G. Casella and R. Berger. *Statistical inference*. Duxbury Press, 2001.

K. Chai. Variational multinomial logit Gaussian process. *The Journal of Machine Learning Research*, 98888:1745–1808, 2012.

E. Challis and D. Barber. Concave Gaussian variational approximations for inference in large-scale Bayesian linear models. In *International conference on Artificial Intelligence and Statistics*, volume 6, page 7, 2011.

S. Chib. Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association*, 90:1313–1321, 1995.

S. Chib and E. Greenberg. Analysis of multivariate probit models. *Biometrika*, 85(2):347–361, 1998.

S. Chib and I. Jeliazkov. Marginal likelihood from the Metropolis-Hastings output. *Journal of the American Statistical Association*, 96(453):270–281, 2001.

W. Chu and Z. Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6:1–48, 2005.

M. Collins, S. Dasgupta, and R. E. Schapire. A generalization of principal components analysis to the exponential family. In *Advances in Neural Information Processing Systems*, 2002.

B. Cseke and T. Heskes. Improving posterior marginal approximations in latent Gaussian models. In *International conference on Artificial Intelligence and Statistics*, volume 9, pages 121–128, 2010.

B. Cseke and T. Heskes. Approximate marginals in latent Gaussian models. *The Journal of Machine Learning Research*, 12:417–454, 2011.

P. Dellaportas and A. F. M. Smith. Bayesian inference for generalized linear and proportional hazards models via Gibbs sampling. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 42(3):pp. 443–459, 1993.

B. Delyon, M. Lavielle, and E. Moulines. Convergence of a stochastic approximation version of the EM algorithm. *Annals of Statistics*, 27(1): 94–128, 1999.

A. Dempster. Covariance selection. *Biometrics*, 28(1), 1972.

S. Duane, A. Kennedy, B. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics letters B*, 195(2):216–222, 1987.

M. Friedlander and M. Schmidt. Hybrid deterministic-stochastic methods for data fitting. *Arxiv preprint arXiv:1104.2373*, 2011.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432, 2008.

S. Frühwirth-Schnatter and R. Frühwirth. Data augmentation and MCMC for binary and multinomial logit models. *Statistical Modelling and Regression Structures*, pages 111–132, 2010.

S. Frühwirth-Schnatter and H. Wagner. Marginal likelihoods for non-Gaussian models using auxiliary mixture sampling. *Computational Statistics and Data Analysis*, 52(10):4608 – 4624, 2008.

D. Gamerman. Sampling from the posterior distribution in generalized linear mixed models. *Statistics and Computing*, 7:57–68, 1997.

S. Gerrish and D. Blei. Predicting legislative roll calls from text. In *Proc. of ICML*, 2011.

Z. Ghahramani and G. Hinton. The EM algorithm for mixtures of factor analyzers. Technical report, Dept. of Comp. Sci., Uni. Toronto, 1996.

M. Girolami and B. Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

M. Girolami and S. Rogers. Variational Bayesian multinomial probit regression with Gaussian process priors. *Neural Comptuation*, 18(8):1790 – 1817, 2006.

Y. Guo and D. Schuurmans. Efficient global optimization for exponential family PCA and low-rank matrix factorization. In *46'th Annual Allerton Conference on Communication, Control, and Computing*, pages 1100–1107. IEEE, 2008.

J. M. Hernández-Lobato and D. Hernández-Lobato. Convergent expectation propogation in linear model with spike-and-slab priors. *arXiv:1112.2289v1*, 2011.

M. Hoffman and A. Gelman. The No-U-Turn Sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *arXiv preprint arXiv:1111.4246*, 2011.

M. Hoffman, D. Blei, and F. Bach. Online learning for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*, 2010.

S. Hoffman and G. Duncan. Multinomial and conditional logit discrete-choice models in demography. *Demography*, 25(3):415–427, 1988.

C. Holmes and L. Held. Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Analysis*, 1(1):145–168, 2006.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 498–520, 1933.

K. Hsiung, S. Kim, and S. Boyd. Tractable approximate robust geometric programming. *Optimization and Engineering*, 9(2):95–118, 2008.

D. R. Hunter and K. Lange. A Tutorial on MM Algorithms. *The American Statistician*, 58:30–37, 2004.

T. Jaakkola. Tutorial on variational approximation methods. In M. Opper and D. Saad, editors, *Advanced mean field methods*. MIT Press, 2001.

T. Jaakkola and M. Jordan. A variational approach to Bayesian logistic regression problems and their extensions. In *International conference on Artificial Intelligence and Statistics*, 1996.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. In M. Jordan, editor, *Learning in Graphical Models*. MIT Press, 1998.

P. Jylänki, J. Vanhatalo, and A. Vehtari. Robust Gaussian process regression with a Student-t likelihood. *The Journal of Machine Learning Research*, 999888:3227–3257, 2011.

A. Kabán and M. Girolami. A combined latent class and trait model for the analysis and visualization of discrete data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):859–872, 2001.

M. E. Khan. An Expectation-Maximization algorithm for learning the latent Gaussian model with Gaussian likelihood. Technical report, Department of Computer Science, University of British Columbia, Vancouver, 2011. URL `http://www.cs.ubc.ca/~emtiyaz/publications.html#Talks`.

M. E. Khan, B. Marlin, G. Bouchard, and K. Murphy. Variational Bounds for Mixed-Data Factor Analysis. In *Advances in Neural Information Processing Systems*, 2010.

M. E. Khan, S. Mohamed, B. Marlin, and K. Murphy. A stick breaking likelihood for categorical data analysis with latent Gaussian models. In *International conference on Artificial Intelligence and Statistics*, 2012a.

M. E. Khan, S. Mohamed, and K. Murphy. Fast Bayesian inference for non-conjugate Gaussian process regression. In *Advances in Neural Information Processing Systems*, 2012b.

S. Kim. A continuation ratio model for ordered category item. In *Annual Meeting of the Psychometric Society*, 2002.

D. Knowles and T. Minka. Non-conjugate variational message passing for multinomial and binary regression. In *Advances in Neural Information Processing Systems*, 2011.

M. Kuss and C. Rasmussen. Assessing approximate inference for binary Gaussian process classification. *Journal of Machine Learning Research*, 6: 1679–1704, 2005.

P. Lenk and W. DeSarbo. Bayesian inference for finite mixtures of generalized linear models with random effects. *Psychometrika*, 65(1):93–119, March 2000.

R. Levine and G. Casella. Implementations of the Monte Carlo EM algorithm. *Journal of Computational and Graphical Statistics*, 10(3):422–439, 2001.

D. MacKay. *Information theory, inference, and learning algorithms*. Cambridge Univesity Press, 2003.

B. Marlin, M. Khan, and K. Murphy. Piecewise bounds for estimating Bernoulli-logistic latent Gaussian models. In *International Conference on Machine Learning*, 2011.

J. Marschak. Binary-choice constraints and random utility indicators. In *Proceedings of a Symposium on Mathematical Methods in the Social Sciences*, 1960.

R. Mazumder and T. Hastie. The graphical lasso: New insights and alternatives. *Arxiv preprint arXiv:1111.5479*, 2011.

P. McCullagh. Regression models for ordinal data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 42(2):109–142, 1980.

P. McCullagh and J. Nelder. *Generalized linear models*. Chapman and Hall, 1989. 2nd edition.

C. McCulloch. Maximum likelihood algorithms for generalized linear mixed models. *Journal of the American statistical Association*, pages 162–170, 1997.

D. McFadden. Conditional logit analysis of qualitative choice behavior. In P. Zarembka, editor, *Frontiers in Econometrics*, pages 105–142. Academic Press, 1973.

T. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2001.

S. Mohamed, K. Heller, and Z. Ghahramani. Bayesian Exponential Family PCA. In *Advances in Neural Information Processing Systems*, 2008.

P. D. Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of Royal Statistical Society, Series B*, 68(3):411–436, 2006.

I. Murray and R. P. Adams. Slice sampling covariance hyperparameters of latent Gaussian models. In *Advances in Neural Information Processing Systems 23*, pages 1723–1731, 2010.

R. Neal. Bayesian learning via stochastic dynamics. In *Advances in Neural Information Processing Systems*, pages 475–482, 1992.

R. Neal. Erroneous results in "Marginal likelihood from the Gibbs output". Technical report, University of Toronto, 1999. URL `http://www.cs.toronto.edu/\textasciitilderadford/chib-letter.html`.

R. M. Neal. Annealed importance sampling. *Statistics and Computing*, 11: 125–139, 2001.

J. Nelder and R. Mead. A simplex method for function minimization. *The computer journal*, 7(4):308, 1965.

M. Newton and A. Raftery. Approximate Bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 3–48, 1994.

H. Nickisch and C. Rasmussen. Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, 9(10), 2008.

M. Opper and C. Archambeau. The variational Gaussian approximation revisited. *Neural computation*, 21(3):786–792, 2009.

S. Orlitsky. Semi-parametric exponential family PCA. *Advances in Neural Information Processing Systems*, 2004.

J. Paisley, D. Blei, and M. Jordan. Variational Bayesian inference with stochastic search. In *International Conference on Machine Learning*, 2012.

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(11):559–572, 1901.

B. Polyak and A. Juditsky. Acceleration of stochastic approximation by averaging. *Siam J. Control Optim*, 30(4):838–855, 1992.

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

M. Rattray, O. Stegle, K. Sharp, and J. Winn. Inference algorithms and learning theory for bayesian sparse factor analysis. In *Journal of Physics: Conference Series*, volume 197, 2009.

P. Rossi and G. Allenby. Bayesian statistics and marketing. *Marketing Science*, 22(3):304–328, 2003.

S. Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, 11(2), 1999.

W. Rudin. *Real and complex analysis*. Tata McGraw-Hill Education, 2006.

H. Rue, S. Martino, and N. Chopin. Approximate Bayesian inference for latent Gaussian models using integrated nested Laplace approximations. *Journal of Royal Statistical Sociecty, Series B*, 71:319–392, 2009.

R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *International Conference on Machine Learning*, pages 880–887. ACM, 2008a.

R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. *Advances in Neural Information Processing Systems*, 20:1257–1264, 2008b.

F. Samejima. Graded response model. *Handbook of modern item response theory*, pages 85–100, 1997.

S. L. Scott. Data augmentation, frequentist estimation, and the Bayesian analysis of multinomial logit models. *Statistical Papers*, 52(1):87–109, 2011.

M. Seeger and M. I. Jordan. Sparse Gaussian process classification with multiple classes. Technical Report Department of Statistics TR 661, University of California, Berkeley, 2004.

M. Seeger and H. Nickisch. Fast convergent algorithms for expectation propagation approximate Bayesian inference. In *International conference on Artificial Intelligence and Statistics*, 2011.

M. Seeger, N. Lawrence, and R. Herbrich. Efficient nonparametric Bayesian modelling with sparse Gaussian process approximations. Technical report, Max Planck Institute, 2006.

A. Skrondal and S. Rabe-Hesketh. *Generalized latent variable modeling: Multilevel, longitudinal, and structural equation models*. CRC Press, 2004.

C. Spearman. "General Intelligence," objectively determined and measured. *The American Journal of Psychology*, 15(2):201–292, 1904.

D. Stern, R. Herbrich, and T. Graepel. Matchbox: large scale online Bayesian recommendations. In *Proceedings of the 18th international conference on World wide web*, pages 111–120. ACM, 2009.

L. Tierney and J. Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, pages 82–86, 1986.

M. Tipping. Probabilistic visualization of high-dimensional binary data. In *Advances in Neural Information Processing Systems*, 1998.

M. Tipping and C. Bishop. Probabilistic principal component analysis. *Journal of Royal Statistical Sociecty, Series B*, 21(3):611–622, 1999.

K. Train. *Discrete choice methods with simulation*. Cambridge University Press, 2003.

D. Van Dyk and X. Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, 2001.

M. Van Gerven, B. Cseke, F. De Lange, and T. Heskes. Efficient Bayesian multivariate fMRI analysis using a sparsifying spatio-temporal prior. *NeuroImage*, 50(1):150–161, 2010.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1–2:1–305, 2008.

M. Wedel and W. Kamakura. Factor analysis with (mixed) observed and latent variables in the exponential family. *Psychometrika*, 66(4):515–530, December 2001.

G. C. G. Wei and M. A. Tanner. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411):699–704, 1990.

M. Welling, C. Chemudugunta, and N. Sutter. Deterministic latent variable models and their pitfalls. In *International Conference on Data Mining*, 2008.

F. Yan and Y. Qi. Sparse Gaussian process regression via l_1 penalization. In *International Conference on Machine Learning*, pages 1183–1190, 2010.

K. Yu, S. Zhu, J. Lafferty, and Y. Gong. Fast nonparametric matrix factorization for large-scale collaborative filtering. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 211–218. ACM, 2009.

S. Zeger and M. Karim. Generalized linear models with random effects: a Gibbs sampling approach. *Journal of the American statistical association*, pages 79–86, 1991.

O. Zoeter, T. Heskes, and B. Kappen. Gaussian quadrature based expectation propagation. In *Workshop on Artificial Intelligence and Statistics*, volume 10, 2005.

A.1 Expectation Identity

Given $q(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{m}, \mathbf{V})$, we prove the following identity:

$$\mathbb{E}_q\left[f(\mathbf{z})\exp(\boldsymbol{\beta}^T\mathbf{z})\right] = e^{\boldsymbol{\beta}^T\mathbf{m}+\frac{1}{2}\boldsymbol{\beta}^T\mathbf{V}\boldsymbol{\beta}}\mathbb{E}_{\tilde{q}}f(\mathbf{z}) \tag{A.1}$$

$$\tilde{q}(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{m}+\mathbf{V}\boldsymbol{\beta}, \mathbf{V}) \tag{A.2}$$

Expanding the left hand side, we get the following,

$$\mathbb{E}_q\left[f(\mathbf{z})\exp(\boldsymbol{\beta}^T\mathbf{z})\right] = \int f(\mathbf{z})e^{\boldsymbol{\beta}^T\mathbf{z}}\frac{1}{|2\pi\mathbf{V}|^{1/2}}e^{-\frac{1}{2}(\mathbf{z}-\mathbf{m})^T\mathbf{V}^{-1}(\mathbf{z}-\mathbf{m})}d\mathbf{z} \tag{A.3}$$

$$= \int f(\mathbf{z})\frac{1}{|2\pi\mathbf{V}|^{1/2}}e^{-\frac{1}{2}(\mathbf{z}-\mathbf{m})^T\mathbf{V}^{-1}(\mathbf{z}-\mathbf{m})+\boldsymbol{\beta}^T\mathbf{z}}d\mathbf{z} \tag{A.4}$$

We complete the square on the exponential term,

$$-\tfrac{1}{2}(\mathbf{z}-\mathbf{m})^T\mathbf{V}^{-1}(\mathbf{z}-\mathbf{m}) + \boldsymbol{\beta}^T\mathbf{z} \tag{A.5}$$

$$= -\tfrac{1}{2}\mathbf{z}^T\mathbf{V}^{-1}\mathbf{z} + \mathbf{m}^T\mathbf{V}^{-1}\mathbf{z} - \tfrac{1}{2}\mathbf{m}^T\mathbf{V}^{-1}\mathbf{m} + \boldsymbol{\beta}^T\mathbf{z} \tag{A.6}$$

$$= -\tfrac{1}{2}\mathbf{z}^T\mathbf{V}^{-1}\mathbf{z} + (\mathbf{m}+\mathbf{V}\boldsymbol{\beta})^T\mathbf{V}^{-1}\mathbf{z} - \tfrac{1}{2}\mathbf{m}^T\mathbf{V}^{-1}\mathbf{m} \tag{A.7}$$

$$= -\tfrac{1}{2}\mathbf{z}^T\mathbf{V}^{-1}\mathbf{z} + (\mathbf{m}+\mathbf{V}\boldsymbol{\beta})^T\mathbf{V}^{-1}\mathbf{z} - \tfrac{1}{2}(\mathbf{V}\boldsymbol{\beta}+\mathbf{m})^T\mathbf{V}^{-1}(\mathbf{V}\boldsymbol{\beta}+\mathbf{m}) + \tfrac{1}{2}(\mathbf{V}\boldsymbol{\beta}+\mathbf{m})^T\mathbf{V}^{-1}(\mathbf{V}\boldsymbol{\beta}+\mathbf{m}) - \tfrac{1}{2}\mathbf{m}^T\mathbf{V}^{-1}\mathbf{m} \tag{A.8}$$

$$= -\tfrac{1}{2}(\mathbf{z}-\mathbf{m}-\mathbf{V}\boldsymbol{\beta})^T\mathbf{V}^{-1}(\mathbf{z}-\mathbf{m}-\mathbf{V}\boldsymbol{\beta}) + \boldsymbol{\beta}^T\mathbf{m} + \tfrac{1}{2}\boldsymbol{\beta}^T\mathbf{V}\boldsymbol{\beta} \tag{A.9}$$

Substituting this back in Eq. A.4, we get the following simplified expression,

$$\mathbb{E}_q\left[f(\mathbf{z})\exp(\boldsymbol{\beta}^T\mathbf{z})\right] = \int f(\mathbf{z})\frac{1}{|2\pi\mathbf{V}|^{1/2}}e^{-\frac{1}{2}(\mathbf{z}-\mathbf{m}-\mathbf{V}\boldsymbol{\beta})^T\mathbf{V}^{-1}(\mathbf{z}-\mathbf{m}-\mathbf{V}\boldsymbol{\beta})} e^{\boldsymbol{\beta}^T\mathbf{m}+\frac{1}{2}\boldsymbol{\beta}^T\mathbf{V}\boldsymbol{\beta}}d\mathbf{z} \tag{A.10}$$

$$= e^{\boldsymbol{\beta}^T\mathbf{m}+\frac{1}{2}\boldsymbol{\beta}^T\mathbf{V}\boldsymbol{\beta}}\int f(\mathbf{z})\mathcal{N}(\mathbf{z}|\mathbf{m}+\mathbf{V}\boldsymbol{\beta}, \mathbf{V})d\mathbf{z} \tag{A.11}$$

$$= e^{\boldsymbol{\beta}^T\mathbf{m}+\frac{1}{2}\boldsymbol{\beta}^T\mathbf{V}\boldsymbol{\beta}}\mathbb{E}_{\tilde{q}}f(\mathbf{z}) \tag{A.12}$$

where $\tilde{q} = \mathcal{N}(\mathbf{z}|\mathbf{m}+\mathbf{V}\boldsymbol{\beta}, \mathbf{V})$.

A.2 Proof of Theorem 3.4.1

To prove concavity with respect to $\boldsymbol{\gamma}_n$, we first prove strict concavity of all the terms except $\underline{f}$. This function is separable in $\mathbf{m}_n$ and $\mathbf{V}_n$, so proving

concavity separately for each variable will establish joint concavity. The function with respect to $\mathbf{m}_n$ is strictly concave since it is a least-square function. Similarly, the function with respect to $\mathbf{V}_n$ is strictly concave since log-det term is strictly concave and addition of a linear trace term maintains the strict concavity (this is because the linear term does not affect the Hessian). Now, back to LVBs. If $\underline{f}(y_{dn}, \widetilde{\boldsymbol{\gamma}}_{dn}, \boldsymbol{\alpha}_{dn})$ is jointly concave with respect to each $\widetilde{\boldsymbol{\gamma}}_{dn}$, then it is also jointly concave with respect to $\boldsymbol{\gamma}_n$ since $\widetilde{\boldsymbol{\gamma}}_{dn}$ are linear functions of $\boldsymbol{\gamma}_n$ (Theorem 3.2.2 of Boyd and Vandenberghe [2004]). Finally, sum of a strictly concave function and a concave function is strictly concave, completing the proof for concavity with respect to $\boldsymbol{\gamma}_n$.

We now prove concavity with respect to each element of $\boldsymbol{\theta}$. Strict concavity with respect to $\boldsymbol{\mu}$ is obvious since the function is a least-squares function. Function with respect to $\boldsymbol{\Sigma}^{-1}$ takes the form of a strictly concave function: $\frac{1}{2}(N \log |\boldsymbol{\Sigma}^{-1}| - \mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}))$ where $\mathbf{S}$ is a symmetric matrix. Concavity with respect to $\mathbf{W}_d$ and $\mathbf{w}_0$ can be established by proving the concavity of a function $f(\widetilde{m}, \widetilde{v})$ with respect to $\mathbf{w}$ and w_0, given that f is jointly concave and that $\widetilde{m} = \mathbf{w}^T\mathbf{m} + w_0$ and $\widetilde{v} = \mathbf{w}^T\mathbf{V}\mathbf{w}$. Concavity with respect to w_0 is obvious since $\widetilde{m}$ is a linear function of w_0. Proving concavity with respect to $\mathbf{w}$ requires a little bit of work since $\widetilde{v}$ is a quadratic function of $\mathbf{w}$. Let $\bigtriangledown^2 f$ be the 2×2 matrix containing derivatives with respect to $\widetilde{m}$ and $\widetilde{v}$, and $\bigtriangledown_w^2 f$ be the Hessian with respect to $\mathbf{w}$. Using the chain rule, we can write the Hessian of f with respect to $\mathbf{w}$ as follows (see Eq. 3.15 in Boyd and Vandenberghe [2004]),

$$\bigtriangledown_w^2 f = \begin{bmatrix} \mathbf{m} & 2\mathbf{V}\mathbf{w} \end{bmatrix} \bigtriangledown^2 f \begin{bmatrix} \mathbf{m}^T \\ 2\mathbf{w}^T\mathbf{V}^T \end{bmatrix} + 2\mathbf{V}\frac{\partial^2 f}{\partial \widetilde{v}^2} \tag{A.13}$$

For concavity, we need this matrix to be negative semi-definite. The last term is negative definite since f is concave with respect to $\widetilde{v}$. First term is a function of the form $\mathbf{T}\mathbf{H}\mathbf{T}^T$ with $\mathbf{H}$ being negative definite. An inner product is always negative or zero, since $\mathbf{x}^T\mathbf{T}\mathbf{H}\mathbf{T}^T\mathbf{x} = \mathbf{y}^T\mathbf{H}\mathbf{y} \leq 0$, proving that the combination is negative semi-definite. This proves that the function is concave with respect to $\mathbf{W}$ and $\mathbf{w}_0$.

A.3 Derivation of the Jaakkola Bound

The Jaakkola bound can be derived using the Fenchel inequality [Boyd and Vandenberghe, 2004]. We first rewrite the LLP function as follows,

$$\mathrm{llp}(\eta) = \log(1 + e^{\eta}) = \eta/2 + \log\left(e^{-\eta/2} + e^{\eta/2}\right) \tag{A.14}$$

Define $x := \eta^2$ and $h(x) := \log\left(e^{-\eta/2} + e^{\eta/2}\right) = \log\left(e^{-\sqrt{x}/2} + e^{\sqrt{x}/2}\right)$. This function is concave in x (which can be verified by taking second derivative), and we get an upper bound using Fenchel's inequality Boyd and Vandenberghe [2004, Chapter 3, Sec. 3.3.2] as shown in Eq. A.15. Here, $h^*(\lambda)$ is the concave conjugate which is defined in Eq. A.16.

$$h(x) \leq \lambda x - h^*(\lambda), \quad \lambda > 0 \tag{A.15}$$

$$h^*(\lambda) = \min_x \lambda x - h(x) \tag{A.16}$$

The value of the conjugate function $h^*(\lambda)$ is not available in closed form, but fortunately it can be parameterized in terms of a minimizer x^* of $\lambda x - h(x)$. By taking the derivative, setting it to zero, and simplifying, we find a condition in Eq. A.19 that x^* should satisfy.

$$\lambda = \left.\frac{\partial h(x)}{\partial x}\right|_{x^*} = \frac{e^{\sqrt{x^*}/2} - e^{-\sqrt{x^*}/2}}{e^{\sqrt{x^*}/2} + e^{-\sqrt{x^*}/2}} \frac{1}{4\sqrt{x^*}} = \frac{1 - e^{-\sqrt{x^*}}}{1 + e^{-\sqrt{x^*}}} \frac{1}{4\sqrt{x^*}} \tag{A.17}$$

$$= \frac{2 - (1 + e^{-\sqrt{x^*}})}{1 + e^{-\sqrt{x^*}}} \frac{1}{4\sqrt{x^*}} = \left(\frac{1}{1 + e^{-\sqrt{x^*}}} - \frac{1}{2}\right) \frac{1}{2\sqrt{x^*}} \tag{A.18}$$

$$= \frac{g\left(\sqrt{x^*}\right) - 1/2}{2\sqrt{x^*}} \tag{A.19}$$

Using x^*, we can rewrite the upper bound as follows,

$$h(x) \leq \lambda x - [\lambda x^* - h(x^*)] = \lambda x - \lambda x^* + \log\left(e^{-\sqrt{x^*}/2} + e^{\sqrt{x^*}/2}\right) \tag{A.20}$$

We use the above to obtain an upper bound to the LLP function. First, we express the LLP function in terms of $h(x)$ using Eq. A.14, as shown in Eq. A.21. We substitute Eq. A.20 to get an upper bound on the LLP function as shown in Eq. A.22. Next, we substitue $x = \eta^2$ and reparameterize $x^* = \xi^2$ to get Eq. A.23, and simplify to get Eq. A.24. The new parameter ξ should be such that $\lambda = \left(g\left(\xi\right) - 1/2\right) / \left(2\xi\right)$, a condition obtained by directly plugging $x^* = \xi^2$ in Eq. A.19.

$$\text{llp}(\eta) = \eta/2 + h(x) \tag{A.21}$$

$$\leq \eta/2 + \lambda x - \lambda x^* + \log\left(e^{-\sqrt{x^*}/2} + e^{\sqrt{x^*}/2}\right) \tag{A.22}$$

$$= \eta/2 + \lambda\eta^2 - \lambda\xi^2 + \log\left(e^{-\xi/2} + e^{\xi/2}\right) \tag{A.23}$$

$$= \eta/2 + \lambda\eta^2 - \lambda\xi^2 - \xi/2 + \text{llp}(\xi) \tag{A.24}$$

Rearraging the terms, we get a quadratic upper bound in Eq. A.25.

$$\text{llp}(\eta) \leq \tfrac{1}{2}a_\xi \eta^2 + \tfrac{1}{2}\eta + c_\xi \tag{A.25}$$

Here, we write λ_ξ to show its dependence on ξ, and other variables are defined as in Eq. 4.7-4.9. We substitute this in Eq. 4.5 and rearrange to obtain the Jaakkola bound shown in Eq. 4.6.

A.4 Derivation of EM algorithm using Quadratic Bounds

We consider the following general quadratic lower bound to the expectation of the log-likelihood,

$$\begin{aligned}\underline{f}^Q(\mathbf{y}_{dn}, \widetilde{\boldsymbol{\gamma}}_{dn}, \boldsymbol{\alpha}_{dn}) = \mathbf{y}_{dn}^T \widetilde{\mathbf{m}}_{dn} - \tfrac{1}{2}\widetilde{\mathbf{m}}_{dn}^T \mathbf{A}_{\alpha,dn} \widetilde{\mathbf{m}}_{dn} + \mathbf{b}_{\alpha,dn} \widetilde{\mathbf{m}}_{dn} - c_{\alpha,dn} \\ - \tfrac{1}{2}\text{Tr}\left(\mathbf{A}_{\alpha,dn} \widetilde{\mathbf{V}}_{dn}\right)\end{aligned} \tag{A.26}$$

The Bohning bound for the Bernoulli logit likelihood is a special case with $\alpha = \psi$ and $a_{\alpha,dn} = 1/4$. The Jaakkola bound is a special with $\alpha = \xi$ and $b_{\alpha,dn} = -1/2$. The Bohning bound for the multinomial logit likelihood is a special case with $\boldsymbol{\alpha} = \boldsymbol{\psi}$ with other functions as defined in Eq. 5.15-5.17. The Gaussian LGM discussed in Section 3.5.2 has similar forms too (although not exactly identical).

We first derive gradients of $\underline{f}^Q$ with respect to $\widetilde{\mathbf{m}}_{dn}$ and $\widetilde{\mathbf{v}}_{dn}$,

$$\mathbf{g}_{dn}^m = \mathbf{y}_{dn} + \mathbf{b}_{\alpha,dn} - \mathbf{A}_{\alpha,dn}\widetilde{\mathbf{m}}_{dn} \quad , \quad \mathbf{G}_{dn}^v = -\tfrac{1}{2}\mathbf{A}_{\alpha,dn} \tag{A.27}$$

We substitute these in the generalized gradient expressions to get the udpates. Denote the vector of $b_{\alpha,dn}$ and $c_{\alpha,dn}$ by $\mathbf{b}_n$ and $\mathbf{c}_n$. We also denote the block diagonal matrix formed with $\mathbf{A}_{\alpha,dn}$ as its diagonal blocks by $\bar{\mathbf{A}}_n$. First, we consider the gradient of $\mathbf{V}_n$ by setting its gradient to 0.

$$\tfrac{1}{2}(\mathbf{V}_n^{-1} - \boldsymbol{\Sigma}^{-1}) + \sum_{d=1}^{D} \mathbf{W}_d \left(-\tfrac{1}{2}\mathbf{A}_{dn}\right) \mathbf{W}_d^T = 0 \tag{A.28}$$

which gives us the update $\mathbf{V}_n = \left(\boldsymbol{\Sigma}^{-1} + \mathbf{W}^T \bar{\mathbf{A}}_n \mathbf{W}\right)^{-1}$. Simlarly, we simplify the gradient for $\mathbf{m}_n$ as follows,

$$-\boldsymbol{\Sigma}^{-1}(\mathbf{m}_n - \boldsymbol{\mu}) + \sum_{d=1}^{D} \mathbf{W}_d^T(\mathbf{y}_{dn} + \mathbf{b}_{\alpha,dn} - \mathbf{A}_{\alpha,dn}\widetilde{\mathbf{m}}_{dn}) \tag{A.29}$$

$$= -\boldsymbol{\Sigma}^{-1}(\mathbf{m}_n - \boldsymbol{\mu}) + \mathbf{W}[\mathbf{y}_n + \mathbf{b}_n - \bar{\mathbf{A}}_n(\mathbf{W}\mathbf{m}_n + \mathbf{w}_0)] \tag{A.30}$$

$$= -(\mathbf{W}^T \bar{\mathbf{A}}_n \mathbf{W} + \boldsymbol{\Sigma}^{-1})\mathbf{m}_n + \mathbf{W}^T(\mathbf{y}_n + \mathbf{b}_n - \bar{\mathbf{A}}_n \mathbf{w}_0) + \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \tag{A.31}$$

Setting this to zero, gives us the update $\mathbf{m}_n = \mathbf{V}_n[\mathbf{W}^T(\mathbf{y}_n + \mathbf{b}_n - \bar{\mathbf{A}}_n\mathbf{w}_0) + \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}]$. Now, we derive the updates for $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{W}, \mathbf{w}_0\}$. Updates with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are available in closed form and are the same as derived earlier in Section 3.5.1. The gradient with respect to $\mathbf{W}_d$ is simplified below,

$$\sum_{n=1}^{N}(\mathbf{y}_{dn} + \mathbf{b}_{\alpha,dn} - \mathbf{A}_{\alpha,dn}\widetilde{\mathbf{m}}_{dn})\mathbf{m}_n^T - \mathbf{A}_{\alpha,dn}\mathbf{W}_d\mathbf{V}_n \tag{A.32}$$

$$= \sum_{n=1}^{N}[\mathbf{y}_{dn} + \mathbf{b}_{\alpha,dn} - \mathbf{A}_{\alpha,dn}(\mathbf{W}_d\mathbf{m}_n + \mathbf{w}_{0d})]\mathbf{m}_n^T - \mathbf{A}_{\alpha,dn}\mathbf{W}_d\mathbf{V}_n \tag{A.33}$$

$$= -\sum_{n=1}^{N}\mathbf{A}_{\alpha,dn}\mathbf{W}_d(\mathbf{m}_n\mathbf{m}_n^T + \mathbf{V}_n) + \sum_{n=1}^{N}(\mathbf{y}_{dn} + \mathbf{b}_{\alpha,dn} - \mathbf{A}_{\alpha,dn}\mathbf{w}_{0d})\mathbf{m}_n^T \tag{A.34}$$

This does not always give rise to simple updates. In case, $\mathbf{A}_{\alpha,dn} = \mathbf{A}$, we get the following update,

$$\mathbf{W}_d = \left[\sum_{n=1}^{N}\left\{\mathbf{A}^{-1}(\mathbf{y}_{dn} + \mathbf{b}_{\alpha,dn}) - \mathbf{w}_{0d}\right\}\mathbf{m}_n^T\right]\left[\sum_{n=1}^{N}\mathbf{V}_n + \mathbf{m}_n\mathbf{m}_n^T\right]^{-1} \tag{A.35}$$

Finally, setting derivative with respect to $\mathbf{w}_{0d}$ to 0, we obtain its update.

$$\sum_{n=1}^{N}\mathbf{y}_{dn} + \mathbf{b}_{dn} - \mathbf{A}_{\alpha,dn}(\mathbf{W}_d\mathbf{m}_n + \mathbf{w}_{0d}) = 0 \tag{A.36}$$

$$\mathbf{w}_{0d} = \left(\sum_{n=1}^{N}\mathbf{A}_{\alpha,dn}\right)^{-1}\sum_{n=1}^{N}(\mathbf{y}_{dn} + \mathbf{b}_{dn} - \mathbf{A}_{\alpha,dn}\mathbf{W}_d\mathbf{m}_n) \tag{A.37}$$

For the Bohning bound, updates for $\mathbf{W}_d$ and $\mathbf{w}_{0d}$ can be written for all d in one matrix form, simplifying the expression further.

A.5 Truncated Gaussian Moments

Given a Gaussian random variable x with mean μ and variance σ^2, we show how to compute

$$f(\mu, \sigma^2, \boldsymbol{\alpha}) = \int_l^h (ax^2 + bx + c)\mathcal{N}(x|\mu, \sigma^2)dx \tag{A.38}$$

and its derivatives with respect to μ and σ^2, where $\boldsymbol{\alpha} = [a, b, c, l, h]$ with all elements of the set being real-valued scalars. We introduce the notation $E_l^h[x^m|\mu, \sigma^2]$ to indicate the truncated expectation $\int_l^h x^m \mathcal{N}(x|\mu, \sigma^2)dx$, where m is a non-negative integer. We can then express $f(\mu, \sigma^2, \boldsymbol{\alpha})$ as $(aE_l^h[x^2|\mu, \sigma^2]+bE_l^h[x^1|\mu, \sigma^2]+cE_l^h[x^0|\mu, \sigma^2])$. The computation of $f(\mu, \sigma^2, \boldsymbol{\alpha})$ and its derivatives then follows from the computation of $E_l^h[x^m|\mu, \sigma^2]$ and its derivatives. We use $\phi(x)$ as shorthand for the standard normal probability density function and $\Phi(x)$ as shorthand for the standard normal cumulative distribution function. We define the standardized variables $\tilde{l} = (l - \mu)/\sigma$ and $\tilde{h} = (h - \mu)/\sigma$. It is easy to see that the computational cost of using a piecewise quadratic bound is only marginally higher than using a piecewise linear bound. This is due to the fact that the Gaussian CDF and PDF functions need only be computed twice each per piece for either class of bounds.

The truncated moments of orders zero, one and two are given below. These moments are closely related to the moments of a truncated and renormalized Gaussian distribution.

$$E_l^h[x^0|\mu, \sigma^2] = \Phi(\tilde{h}) - \Phi(\tilde{l}) \tag{A.39}$$

$$E_l^h[x^1|\mu, \sigma^2] = \sigma(\phi(\tilde{l}) - \phi(\tilde{h})) + \mu(\Phi(\tilde{h}) - \Phi(\tilde{l})) \tag{A.40}$$

$$E_l^h[x^2|\mu, \sigma^2] = \sigma^2(\tilde{l}\phi(\tilde{l}) - \tilde{h}\phi(\tilde{h})) + (\sigma^2 + \mu^2)(\Phi(\tilde{h}) - \Phi(\tilde{l})) \tag{A.41}$$

The derivatives of the standard Gaussian PDF and CDF evaluated at $\tilde{x}$ are given below, which are both implicit function of μ and σ due to the definition $\tilde{x} = (x - \mu)/\sigma$.

$$\frac{\partial\phi(\tilde{x})}{\partial\mu} = \frac{\tilde{x}}{\sigma}\phi(\tilde{x}) \quad , \qquad \frac{\partial\phi(\tilde{x})}{\partial\sigma^2} = \frac{\tilde{x}^2}{2\sigma^2}\phi(\tilde{x}) \tag{A.42}$$

$$\frac{\partial\Phi(\tilde{x})}{\partial\mu} = -\frac{1}{\sigma}\phi(\tilde{x}) \quad , \qquad \frac{\partial\Phi(\tilde{x})}{\partial\sigma^2} = -\frac{\tilde{x}}{2\sigma^2}\phi(\tilde{x}) \tag{A.43}$$

Using these, the derivatives of each truncated moment $E_l^h[x^m|\mu, \sigma^2]$ with respect to μ and σ^2 can be computed and are given below. These are all the derivatives needed to compute the gradients of the piecewise bound.

$$\frac{\partial E_l^h[x^0|\mu, \sigma^2]}{\partial\mu} = \frac{1}{\sigma^2}\left(\phi(\tilde{l}) - \phi(\tilde{h})\right) \tag{A.44}$$

$$\frac{\partial E_l^h[x^0|\mu, \sigma^2]}{\partial\sigma^2} = \frac{1}{2\sigma^2}\left(\tilde{l}\phi(\tilde{l}) - \tilde{h}\phi(\tilde{h})\right) \tag{A.45}$$

$$\frac{\partial E_l^h[x^1|\mu,\sigma^2]}{\partial \mu} = \frac{1}{\sigma}\left(l\phi(\tilde{l}) - h\phi(\tilde{h})\right) + \Phi(\tilde{h}) - \Phi(\tilde{l}) \tag{A.46}$$

$$\frac{\partial E_l^h[x^1|\mu,\sigma^2]}{\partial \sigma^2} = \frac{l^2+\sigma^2-l\mu}{2\sigma^3}\phi(\tilde{l}) - \frac{h^2+\sigma^2-h\mu}{2\sigma^3}\phi(\tilde{h}) \tag{A.47}$$

$$\begin{aligned}\frac{\partial E_l^h[x^2|\mu,\sigma]}{\partial \mu} = \frac{1}{\sigma}&\left((l^2+2\sigma^2)\phi(\tilde{l}) - (h^2+2\sigma^2)\phi(\tilde{h})\right) \\ &+ 2\mu\left(\Phi(\tilde{h}) - \Phi(\tilde{l})\right)\end{aligned} \tag{A.48}$$

$$\begin{aligned}\frac{\partial E_l^h[x^2|\mu,\sigma^2]}{\partial \sigma^2} = (l^3+2\sigma^2 l - l^2\mu)&\frac{\phi(\tilde{l})}{2\sigma^3} - (h^3+2\sigma^2 h - h^2\mu)\frac{\phi(\tilde{h})}{2\sigma^3} \\ &+ \Phi(\tilde{h}) - \Phi(\tilde{l})\end{aligned} \tag{A.49}$$

A.6 Derivation of the Log Bound

We use the following inequality, $\log x \leq \nu x - \log\nu - 1$ where $\nu > 0$. The above inequality can be obtained using the Fenchel's inequality (see Boyd and Vandenberghe [2004, Ch. 3, Sec. 3.3.2]). Using this inequality, an upper bound on the LSE function is obtained as defined in Eq. A.50.

$$\text{lse}(\boldsymbol{\eta}) \leq \nu \sum_{k=0}^{K} e^{\eta_k} - \log\nu - 1 \tag{A.50}$$

Here $\nu > 0$ is the local variational parameter and needs to be optimized to get a tight bound. Taking expectation with respect to $q(\boldsymbol{\eta}|\widetilde{\boldsymbol{\gamma}})$ and using the identity of Appendix A.1, we obtain the following upper bound on the expectation of the LSE function,

$$\mathbb{E}_{q(\boldsymbol{\eta}|\widetilde{\boldsymbol{\gamma}})}[\text{lse}(\boldsymbol{\eta})] \leq \nu \sum_{k=0}^{K} e^{\widetilde{m}_k + \widetilde{v}_k/2} - \log\nu - 1 \tag{A.51}$$

We can remove the local variational parameter by maximizing with respect to it. Taking derivative of the left hand side with respect to ν and setting it to zero, gives us the optimal choice of $\nu^* = 1/\sum_k \exp(\widetilde{m}_k + \widetilde{v}_k/2)$. Substituting this in the bound, we get the final log bound in Eq. 5.6.

The log bound can also be derived using the zeroth-order delta method. A zeroth-order approximation of a function $f(\boldsymbol{\eta})$ is obtained by taking expectation of a zeroth-order Taylor expansion around the mean $\widetilde{\mathbf{m}}$, as shown below. Here, $\mathbf{g}_f$ is the gradient of function f.

$$\mathbb{E}_{q(\boldsymbol{\eta}|\widetilde{\boldsymbol{\gamma}})}[f(\boldsymbol{\eta})] \approx \mathbb{E}_{q(\boldsymbol{\eta}|\widetilde{\boldsymbol{\gamma}})}[f(\widetilde{\mathbf{m}}) + (\boldsymbol{\eta} - \widetilde{\mathbf{m}})^T \mathbf{g}_f(\widetilde{\mathbf{m}})] = f(\widetilde{\mathbf{m}}) \tag{A.52}$$

This can also be interpreted as "pushing" the expectation inside the function. We know from Jensen's inequality that this results in an upper (lower) bound if the function is concave (convex), i.e. $\mathbb{E}[f(\boldsymbol{\eta})] \leq f(\mathbb{E}(\boldsymbol{\eta}))$. We choose log to be the function h and apply the zeroth-order delta method to obtain the bound.

$$\mathbb{E}_{q(\boldsymbol{\eta}|\widetilde{\boldsymbol{\gamma}})}[\text{lse}(\boldsymbol{\eta})] \leq \log \sum_{k=1}^{K} \mathbb{E}_{q(\eta_k|\widetilde{\gamma}_k)} e^{\eta_k} = \log \sum_{k=1}^{K} e^{\widetilde{m}_k + \widetilde{v}_k/2} \tag{A.53}$$

A.7 Derivation of the Tilted Bound

We start with the expectation of the LSE function in Eq. A.54. Here, the expectation is with respect to $q(\boldsymbol{\eta}|\widetilde{\boldsymbol{\gamma}}) = \mathcal{N}(\boldsymbol{\eta}|\widetilde{\mathbf{m}}, \widetilde{\mathbf{V}})$. We multiply and divide by an exponential term shown in Eq. A.56 and simplify in Eq. A.57.

$$\mathbb{E}[\text{lse}(\boldsymbol{\eta})] = \mathbb{E}\left[\log \sum_{k=0}^{K} e^{\eta_k}\right] \tag{A.54}$$

$$= \mathbb{E}\left[\log \sum_{k=0}^{K} e^{\sum_{j=0}^{K} a_j \eta_j} e^{-\sum_{j=0}^{K} a_j \eta_j} e^{\eta_k}\right] \tag{A.55}$$

$$= \sum_{j=0}^{K} a_j \widetilde{m}_j + \mathbb{E}\left[\log \sum_{k=0}^{K} e^{\eta_k - \sum_{j=0}^{K} a_j \eta_j}\right] \tag{A.56}$$

Define $\boldsymbol{\beta}_k$ as a vector with k'th element as $1 - a_k$ and j'th element as $-a_j$ for $j \neq k$. Then, Eq. A.56 can be rewritten as Eq. A.57. We use Jensen't inequality to take the expectation inside log, to get the upper bound of Eq. A.58. Next, we use the expectation identity of Appendix A.1 to get Eq. A.59.

$$\mathbb{E}[\text{lse}(\boldsymbol{\eta})] = \sum_{j=0}^{K} a_j \widetilde{m}_j + \mathbb{E}\left[\log \sum_{k=0}^{K} \exp\left(\boldsymbol{\beta}_k^T \boldsymbol{\eta}\right)\right] \tag{A.57}$$

$$\leq \sum_{j=0}^{K} a_j \widetilde{m}_j + \log \sum_{k=0}^{K} \mathbb{E}\left[\exp\left(\boldsymbol{\beta}_k^T \boldsymbol{\eta}\right)\right] \tag{A.58}$$

$$= \sum_{j=0}^{K} a_j \widetilde{m}_j + \log \sum_{k=0}^{K} \exp\left(\boldsymbol{\beta}_k^T \widetilde{\mathbf{m}} + \boldsymbol{\beta}_k^T \widetilde{\mathbf{V}} \boldsymbol{\beta}_k / 2\right) \tag{A.59}$$

This lower bound depends on the off-diagonal elements of $\widetilde{\mathbf{V}}$, however this can be expressed in terms of the diagonal elements only, as shown in A.60.

We obtain this bound since $\widetilde{\mathbf{V}}$ is positive definite. Next, we substitute the definition of $\boldsymbol{\beta}_k$ and simplify to bring the common terms out of the LSE term to get the upper bound shown in Eq. A.62.

$$\mathbb{E}[\text{lse}(\boldsymbol{\eta})] \leq \sum_{j=0}^{K} a_j \widetilde{m}_j + \log \sum_{k=0}^{K} \exp\left(\boldsymbol{\beta}_k^T \widetilde{\mathbf{m}} + \boldsymbol{\beta}_k^T \text{diag}(\widetilde{\mathbf{V}})\boldsymbol{\beta}_k/2\right) \tag{A.60}$$

$$= \sum_{j=0}^{K} a_j \widetilde{m}_j + \log \sum_{k=0}^{K} e^{\widetilde{m}_k + (1-2a_j)\widetilde{v}_k/2 + \left(\sum_{j=0}^{K} - a_j \widetilde{m}_j + a_j^2 \widetilde{v}_j/2\right)} \tag{A.61}$$

$$= \tfrac{1}{2}\sum_{j=0}^{K} a_j^2 \widetilde{v}_j + \log \sum_{k=0}^{K} e^{\widetilde{m}_k + (1-2a_j)\widetilde{v}_k/2} \tag{A.62}$$

We see that there are two sources of error in the bound. The first one is very similar to the zeroth-order delta method, and second one is by using only the diagonal elements of $\widetilde{\mathbf{V}}$. The former is more serious and making the bound accurate locally, not ensuring the global tightness.

A.8 Proof of Theorem 5.5.1

The log bound and the "optimal" Bohning bound are shown below,

$$\underline{f}^B(\mathbf{y}, \widetilde{\boldsymbol{\gamma}}, \boldsymbol{\psi}^*) = \mathbf{y}^T \widetilde{\mathbf{m}} - \text{lse}(\widetilde{\mathbf{m}}) - \tfrac{1}{2}\text{Tr}(\mathbf{A}\widetilde{\mathbf{V}}) \tag{A.63}$$

$$\underline{f}^L(\boldsymbol{\theta}, \widetilde{\boldsymbol{\gamma}}) = \mathbf{y}^T \widetilde{\mathbf{m}} - \text{lse}(\widetilde{\mathbf{m}} + \widetilde{\mathbf{v}}/2) \tag{A.64}$$

Using these, we get the following expression for Δ,

$$\Delta = \text{lse}(\widetilde{\mathbf{m}}) + \tfrac{1}{2}\text{Tr}(\mathbf{A}\widetilde{\mathbf{V}}) - \text{lse}(\widetilde{\mathbf{m}} + \widetilde{\mathbf{v}}/2) \tag{A.65}$$

We now find an upper bound on the last term in terms of the first term,

$$\text{lse}(\widetilde{\mathbf{m}} + \widetilde{\mathbf{v}}/2) \leq \log e^{\widetilde{v}_{max}/2} \sum_{k=0}^{K} e^{\widetilde{m}_k + \widetilde{v}_k/2} = \tfrac{1}{2}\widetilde{v}_{\max} + \text{lse}(\widetilde{\mathbf{m}}) \tag{A.66}$$

Similarly, $\text{lse}(\widetilde{\mathbf{m}} + \widetilde{\mathbf{v}}/2) \geq \frac{1}{2}\widetilde{v}_{\min} + \text{lse}(\widetilde{\mathbf{m}})$. Substituting these in Eq. A.65, we get the results.