Variational Bayesian Inference for Big Data Marketing Models

Asim Ansari        Yang Li        Jonathan Z. Zhang

December 2014

1This is a preliminary version. Please do not cite or circulate.
2Asim Ansari is the William T. Dillard Professor of Marketing at Columbia Business School, Yang Li is Assistant Professor of Marketing at Cheung Kong Graduate School of Business in Beijing, and Jonathan Z. Zhang is Assistant Professor of Marketing at University of Washington in Seattle.
Abstract

Hierarchical Bayesian approaches play a central role in empirical marketing as they yield individual-level parameter estimates that can be used for targeting decisions. MCMC methods have been the methods of choice for estimating hierarchical Bayesian models as they are capable of providing accurate individual-level estimates. However, MCMC methods are computationally prohibitive and do not scale well when applied to massive data sets that have become common in the current era of “Big Data”. We introduce to the marketing literature a new class of Bayesian estimation techniques known as variational Bayesian (VB) inference. These methods tackle the scalability challenge via a deterministic optimization approach to approximate the posterior distribution and yield accurate estimates at a fraction of the computational cost associated with simulation-based MCMC methods. We exploit and extend recent developments in variational Bayesian inference and highlight how two VB estimation approaches – Mean-field VB (that is analogous to Gibbs sampling) for conjugate models and Fixed-form VB (which is analogous to Metropolis-Hasting) for non-conjugate models – can be effectively combined for estimating complex marketing models. We also show how recent advances in parallel computing and in stochastic optimization can be used to further enhance the speed of these VB methods. Using simulated as well as real data sets, we apply the VB approaches to several commonly used marketing models (e.g. mixed linear, logit, selection, and hierarchical ordinal logit models), and demonstrate how the VB inference is widely applicable for marketing problems.

Keywords: Big Data, Variational Inference, Mean-field Variational Bayes, Fixed-form Variational Bayes, Parallelization, Stochastic Optimization, Adaptive Mini-batch, MovieLens Database.
1 Introduction

The recent advances in information technology, coupled with the rapid decrease in data storage cost, have resulted in the explosive growth in the collection and availability of customer level data. Consumer interactions governed by the Internet, e-commerce, social media, geographical positioning systems and information-sensing mobile devices result in a massive trail of data-points, collected at a pace faster than ever before (Boyd and Crawford 2012). As the volume of business data worldwide, across all companies, doubles every 1.2 years (eMarketer 2013), data set sizes in empirical marketing research have also increased significantly over the years. For example, within the past decade or so, in marketing papers using scanner data, we have seen data set sizes increase from the thousands (e.g., Villas-Boas and Winer 1999) to hundreds of thousands of observations (e.g., Gordon, Goldfarb and Li 2013).

These novel phenomena, collectively known as “Big Data”, provide scholars with exciting opportunities to understand, explain, and predict the behavior of consumers in more detailed ways than before. Effective analyses of such large customer databases can yield insights on individual preferences and can help firms increase profit via better targeting, such as in the cases of Harrah’s Entertainment, Capital One, and Netflix (Davenport and Harris 2007). However, this data deluge also challenges firms’ ability to process data in an effective and managerially timely manner.

Big Data is characterized by high volume, high velocity, and high variety that “require new forms of processing to enable enhanced decision making, insight discovery and process optimization” (Gartner 2012). In this research, of the “3Vs of Big Data”, we tackle the issues of high data volume and velocity by introducing to the marketing literature a new and rapidly evolving Bayesian estimation framework, known as Variational Bayesian (VB) inference. This framework is deterministic, scales well in the presence of large number of observations, and allows marketers to expeditiously estimate individual-level response parameters for marketing actions, when other statistical methods would take too long to yield results.

Understanding heterogeneity in consumer preferences is of paramount importance in many marketing activities. Strategically, an understanding of the distribution of consumers responses to product attributes would guide product design decisions – an insight that would be lost if the preference is examined only at the mean (Allenby and Rossi 1999). Tactically, modeling individual-level responses to marketing actions allows firms to adjust allocation of resources across regions, stores, and consumers (Rossi et al. 1996). In the past 20 years, the hierarchical Bayesian methods have become popular in marketing because of their demonstrated
utility in yielding highly accurate individual level estimates by appropriately pooling information across consumers. On the estimation front, advances in Markov chain Monte Carlo methods have fueled this Bayesian revolution in marketing (Rossi, Allenby and McCulloch 2005).

The marketing literature has used a variety of MCMC methods for estimating hierarchical Bayesian models. For conjugate models, Gibbs sampling (Gelfand and Smith 1990; Gelfand et al. 1990) is the method of choice. In contrast, for non-conjugate models, the full conditional distributions do not have closed-form, and Metropolis-Hasting methods (Chib and Greenberg 1995) and their extensions such as Hamiltonian MCMC and slice sampling (Neal 2003; Neal 2011), have been used across a wide spectrum of applications. Streams of research that leverage individual-level information have flourished, to address a wide array of topics such as couponing (Rossi et al. 1996), recommendation systems (Ansari, Essegaier and Kohli 2000), digital marketing campaigns (Ansari and Mela 2003), B2B communication contacts (Venkatesan and Kumar 2004), pharmaceutical detailing and sampling (Dong et al. 2009; Montoya et al. 2010), targeted promotions (Zhang and Krishnamurthi 2004), price endogeneity (Li and Ansari 2014), and dynamic targeted pricing (Zhang et al. 2014).

Despite their unquestionable usefulness and prevalence in marketing, MCMC methods are hampered by the bottlenecks of speed and scalability, and are, therefore, difficult to apply in large data situations where data sets can often contain hundreds of thousands or even millions of individuals. The situation is further exacerbated by models with complex posteriors such as those involving non-conjugacy, multiple decisions, and dynamics. It is ironic that while large data sets contain enough information content to support complex models without the risk of overfitting, estimating these models becomes intractable using sampling based methods. MCMC methods suffer from poor mixing, especially in hierarchical models involving a large number of latent variables, and therefore, require a large number of parameter draws for convergence. Mixing also becomes difficult in large data situations because of a concentrated posterior distribution. It is therefore not uncommon in such situations for an MCMC simulation to run for days or even weeks to ensure convergence and to obtain a reasonable number of effectively uncorrelated samples. In short, in the face of big data, traditional MCMC methods do not scale well and converge too slowly to be useful for making timely managerial decisions including recalibrating price sensitivities for customers, making product recommendations, or customizing advertising by firms such as Netflix, Amazon, Google, and those involved in mobile targeting.

To address the scalability issue, it is common to prune the data and only work with a subset of individuals
or a subset of observations. This approach, however, discards information that is valuable for estimation and for targeting to every individual, and might lead to poor estimates (Zanutto and Bradlow 2006). Other methods of data compression, for instance discarding old data, the judicious choice of sufficient statistics, data aggregation (Musalem, Bradlow and Raju 2009) and data squashing (Madigan et al. 2002) can also be used.

In this paper, we show that variational Bayesian approaches can be used as an efficient and scalable alternative to MCMC methods in such “Big Data” situations. In contrast to MCMC methods that simulate from the posterior, variational Bayesian methods use an optimization approach to construct an approximation to the posterior. They offer the benefit of significant speedups when compared to MCMC methods and can recover parameter estimates accurately. We derive variational algorithms for the conjugate (linear) and non-conjugate (logit and ordinal logit) hierarchical models commonly used in marketing, and illustrate how different variational methods that are counterparts to Gibbs and Metropolis-Hasting sampling methods can be used. We also show how recent advances in parallel computing and in stochastic approximation can be used to further enhance the speed of these VB methods. We apply the methods to simulated and real data sets of different sizes and demonstrate that VB methods not only can achieve the same level of accuracy as MCMC, but with speeds that can be up to thousands of times faster than MCMC. We also make available a suite of computer programs that marketing academics and practitioners can readily use to conduct VB inference on different models.

The rest of the paper proceeds as follows. We begin in Section 2 by explaining the basic concept behind variational Bayesian inference. Then in Section 3, we introduce mean-field VB for conjugate models and fixed-form VB for non-conjugate models, and illustrate algorithms for linear-mixed, selection, and logit models. Section 4 shows how mean-field and fixed-form can be combined into a “hybrid” VB to efficiently estimate models with both conjugate and non-conjugate components, and offers an algorithm for one such case - a hierarchical ordinal logit model. In Section 5, we discuss how VB can be further enhanced via parallelization and stochastic optimization with adaptive mini-batch sizes. Section 6 and 7 contain applications of these VB methods and extensions to the hierarchical ordinal logit model. Lastly in Section 8, we summarize the benefits of VB in marketing and highlight potential avenues for future research. All other details are located in the Appendix. The codes and data sets are available from the authors upon request.
2 Variational Bayesian Inference

Bayesian inference is based on summarizing the posterior distribution of all unknowns. In a generic Bayesian model with observed data $y$ and the unknown parameter vector $\theta$, the posterior distribution $p(\theta|y)$ follows the Bayes rule,

$$
p(\theta|y) = \frac{p(y, \theta)}{p(y)} = \frac{p(y|\theta)p(\theta)}{p(y)},
$$

where, $p(\theta)$ is the prior distribution, $p(y|\theta)$ is the likelihood, and $p(y)$ is the normalizing constant or evidence. For almost all interesting marketing models, the normalizing constant cannot be computed in closed-form. The posterior distribution, therefore, is not available analytically. This necessitates the use of approximations to summarize the posterior distribution. Simulation methods based on Markov Chain Monte Carlo (MCMC) are the methods of choice in this regard, given their excellent approximation capabilities. However, MCMC methods can be very time consuming, especially in Big data situations.

In contrast to MCMC methods that approximate the posterior by simulating random draws, variational inference seeks to approximate the intractable posterior, $p(\theta|y)$, with a simpler distribution, $q(\theta|\eta)$, called the variational distribution (Bishop 2006; Ormerod and Wand 2010). The variational distribution belongs to a family of distributions that is indexed by a set of free parameters $\eta$. In variational Bayesian inference, one searches over the space of the free parameters to find a member of the variational family that is closest to the posterior of interest. In short, variational methods recast the model inference problem as an optimization problem, thus making it possible to obtain speed and scalability.

VB methods rely on the Kullback-Leibler (KL) divergence (Kullback and Leibler 1951) to measure the dissimilarity (distance) between two probability distributions.\(^{1}\) The KL divergence between the approximating distribution $q(\theta)$ and the posterior $p(\theta|y)$ is defined as

$$
\text{KL}[q(\theta) || p(\theta|y)] = \int q(\theta) \log \frac{q(\theta)}{p(\theta|y)} d\theta \geq 0,
$$

where, the equality holds if and only if $q(\theta) = p(\theta|y)$ almost everywhere. We can manipulate the KL

\(^{1}\)Note that Euclidean distance between distributional parameters is often a poor measure of closeness between probability distributions (Hoffman et al. 2013). For instance, the two normal distributions $N(0, 100^2)$ and $N(10, 100^2)$ are almost indistinguishable, and the Euclidean distance between their parameter vectors is 10. In contrast, the distributions $N(0, 0.1^2)$ and $N(0.1, 0.1^2)$ barely overlap, but this is not reflected in the Euclidean distance that is only 0.1.
divergence such that

\[
\text{KL} \left[ q(\theta) \mid \mid p(\theta \mid y) \right] = E_q[\log q(\theta)] - E_q[\log p(\theta \mid y)] \\
= E_q[\log q(\theta)] - E_q[\log p(y, \theta)] + \log p(y).
\]

In the above, the expectation \( E_q[\cdot] \) is with respect to the variational distribution \( q(\theta) \), and the last term, \( \log p(y) \) is a constant. Minimizing the KL divergence is thus equivalent to maximizing the scalar quantity,

\[
\mathcal{L}(q) = E_q[\log p(y, \theta)] - E_q[\log q(\theta)],
\]

which is usually referred as the evidence lower bound (ELBO; Bishop 2006; Ormerod and Wand 2010).

Our goal here is to find an approximating variational distribution \( q(\theta) \) that makes KL as close to zero as possible. However, because the posterior is unknown to begin with, we need to impose restrictions on the approximating variational distribution for inference to proceed. These restrictions serve to structure the approximating distribution such that its functional form can be either inferred or set. The machine learning literature has explored a number of different approaches for structuring these approximating distributions. In the current paper, we focus on two approaches that can be effectively applied for marketing models – mean-field approximations for conjugate models and fixed-form approximations for non-conjugate ones. These two approaches can also be used in tandem to yield a hybrid approximation for models that involve both conjugate and non-conjugate elements.

3 Mean-field and Fixed-form Methods

3.1 Mean-field Variational Bayes (MFVB)

The mean-field\(^2\) approximation can be considered the deterministic counterpart to Gibbs sampling, and as in Gibbs sampling, it is applicable for conjugate or semi-conjugate models (Ormerod and Wand 2010; Grimmer 2010). In mean-field inference, the variational distribution \( q(\theta) \) is restricted to a factorized product form \( \prod_{i=1}^{D} q_i(\theta_i) \), over some partition \( \{\theta_1, \ldots, \theta_D\} \) of \( \theta \). Underlying this restriction is the assumption of independence across the different parameter blocks. Such an approximation is nonparametric in spirit. Note that no parametric assumptions regarding the functional form of \( q_i(\theta_i) \) are used. Rather, the optimal functional form, as well as the optimal parameters of the variational distribution can be inferred given only

\(^2\)The name “mean-field” originated from physics.
the likelihood and the prior. Under the above assumption, setting $\partial L(q)/\partial q = 0$ leads to the following optimal solution to the minimization of the Kullback-Leibler divergence (Murphy 2012),

$$q_i^*(\theta_i) \propto \exp\{\mathbb{E}_{\theta_{-i}}[\log p(\theta_i|y, \theta_{-i})]\},$$

(5)

where, $\mathbb{E}_{\theta_{-i}}[\cdot]$ refers to the expectation with respect to the variational distribution of the other parameter blocks, $\theta_{-i}$, and $p(\theta_i|y, \theta_{-i})$ is the posterior full conditional distribution for $\theta_i$. When a conjugate prior is used, the optimal variational component, $q_i^*(\theta_i)$, is available in closed-form and belongs to the same distributional family as the posterior full conditional. Also, because the posterior full conditional exists analytically, the functional form of the variational distribution is readily available.

Denote $q_i(\theta_i) = q_i(\theta_i|\eta_i)$, and $\eta_i$ as the parameter for the $i$th variational distribution. Then finding the optimal variational density only requires optimization to obtain the variational parameters $\{\eta_i\}_{\forall i}$. As we see in the next section, this can be done using simple coordinate ascent optimization in which the different variational parameters are updated sequentially in an iterative and deterministic fashion, until convergence is achieved.

### 3.1.1 A Linear Regression Model

We now illustrate the mean-field approximation and its use in variational Bayesian inference using the most familiar of statistical settings – a linear regression model. Consider

$$y_i = x_i'\beta + \epsilon_i,$$

(6)

$$\epsilon_i \sim N(0, \sigma^2), \quad i = 1 \ldots N,$$

with semi-conjugate priors,

$$\beta \sim N(\mu_\beta, \Sigma_\beta) \quad \text{and} \quad \sigma^2 \sim IG(a, b).$$

While each of the two priors are individually conjugate to the normal likelihood, given the other parameter, the joint prior is not conjugate, and thus the resulting posterior distribution for the model is not tractable. However, one can use a mean-field approach and approximate the posterior using the factorization:

$$q(\beta, \sigma^2) = q(\beta) q(\sigma^2).$$

(7)
According to (5), the optimal densities take the form
\[
q^*(\beta) \propto \exp \mathbb{E}_{q(\sigma^2)}[\log p(\beta|\sigma^2, \mathbf{X}, \mathbf{Y})]
\]
\[
q^*(\sigma^2) \propto \exp \mathbb{E}_{q(\beta)}[\log p(\sigma^2|\beta, \mathbf{X}, \mathbf{Y})].
\]

Given the semi-conjugate setup, the full conditional distributions, \(p(\beta|\sigma^2, \mathbf{X}, \mathbf{Y})\) and \(p(\sigma^2|\beta, \mathbf{X}, \mathbf{Y})\), are fully known. Plugging in these full conditional distributions, and after some algebraic manipulations, one can deduce that \(q^*(\beta)\) is a multivariate normal density \(N(\mu_{q(\beta)}, \Sigma_{q(\beta)})\), where,
\[
\Sigma_{q(\beta)}^{-1} = \Sigma_{\beta}^{-1} + \mathbb{E}_{q(\sigma^2)}[\sigma^{-2}] \mathbf{X}' \mathbf{X},
\]
\[
\mu_{q(\beta)} = \Sigma_{q(\beta)}(\Sigma_{\beta}^{-1} \mu_{\beta} + \mathbb{E}_{q(\sigma^2)}[\sigma^{-2}] \mathbf{X}' \mathbf{Y}),
\]
and \(q^*(\sigma^2)\) is an inverse Gamma density \(\text{IG}(a_{q(\sigma^2)}, b_{q(\sigma^2)})\) with
\[
a_{q(\sigma^2)} = a + N/2,
\]
\[
b_{q(\sigma^2)} = b + 1/2 \sum_{i=1}^{N} \left( (y_i - \mathbf{x}_i' \mu_{q(\beta)})^2 + \text{Tr}[\mathbf{x}_i \mathbf{x}_i' \text{Var}_{q(\beta)}[\beta]] \right),
\]
where \(\text{Tr}[\cdot]\) denotes the trace of a matrix.

It is easy to discern from the above sets of equations that the variational parameters are interdependent. For instance, in (9), the precision of the variational approximation for \(\beta, \Sigma_{q(\beta)}^{-1}\), depends on the mean of the variational distribution for \(\sigma^{-2}, \mathbb{E}_{q(\sigma^2)}[\sigma^{-2}]\). We can therefore sequentially update these variational parameter values (9) and (10) in an iterative fashion. This is similar to Gibbs sampling where we iteratively draw from the full conditionals, except that in variational Bayes, we update the variational parameters directly, instead of sampling the model parameters.

\begin{table}[h]
\centering
\begin{tabular}{|c|}
\hline
\textbf{Algorithm 1}. MFVB for the Linear Regression Model  \\
\hline
1. Initialize \(\Sigma_{q(\beta)}, \mu_{q(\beta)}\) and \(b_{q(\sigma^2)}\).  \\
2. Iteratively update  \\
\(\Sigma_{q(\beta)}^{-1} \leftarrow \Sigma_{\beta}^{-1} + a_{q(\sigma^2)}/b_{q(\sigma^2)} \mathbf{X}' \mathbf{X}\)  \\
\(\mu_{q(\beta)} \leftarrow \Sigma_{q(\beta)}(\Sigma_{\beta}^{-1} \mu_{\beta} + a_{q(\sigma^2)}/b_{q(\sigma^2)} \mathbf{X}' \mathbf{Y})\)  \\
\(b_{q(\sigma^2)} \leftarrow b + 1/2 \sum_{i=1}^{N} \left( (y_i - \mathbf{x}_i' \mu_{q(\beta)})^2 + \text{Tr}[\mathbf{x}_i \mathbf{x}_i' \text{Var}_{q(\beta)}[\beta]] \right)\)  \\
until the increase in ELBO is negligible.  \\
\hline
\end{tabular}
\end{table}
Algorithm 1 describes the resulting coordinate ascent schedule for obtaining the optimal values of the variational parameters for the linear regression model. In the algorithm, ELBO is derived from (4) and given by

\[
\mathcal{L}(q) = - (\mu_{q(\beta)} - \mu_\beta)' \Sigma_\beta^{-1} (\mu_{q(\beta)} - \mu_\beta) - \text{Tr}[\Sigma_\beta^{-1} \Sigma_{q(\beta)}] \\
+ \log |\Sigma_{q(\beta)}| - 2a_{q(\sigma^2)} \log b_{q(\sigma^2)}.
\]

The preceding discussion serves as a tutorial for explaining the basics of applying mean-field approximation in a simplified setting. However, this is too simple a model to fully illustrate the effectiveness of variational Bayesian methods. We now show the use of VB in more complicated hierarchical settings. In the next section, we discuss how to use mean-field methods for a cross-nested mixed linear model, which has wide applicability in marketing research.

3.1.2 A Cross-Nested Mixed Linear Model

Marketing research environments are replete with panel data which require careful modeling of multiple sources of unobserved heterogeneity (Allenby and Rossi 1999). In many settings, data are available on multiple consumers and on many different products. For instance, data from recommender systems include ratings from different users on many different items. A proper accounting of the variation in such data sets requires the use of random effects for products as well as for customers (Ansari, Essegaeir and Kohli 2000), resulting in a cross-nested structure.

Here we consider a linear model with cross-nested random coefficients (Rasbash and Browne 2008) to account for both individual as well as product heterogeneity. Specifically, we simulate panel data sets from the following model,

\[
y_{ij} = x'_{ij}\beta + z'_j\lambda_i + w'_i\gamma_j + \epsilon_{ij}, \\
\epsilon_{ij} \sim N(0, \sigma^2), \quad \lambda_i \sim N(0, \Lambda), \quad \gamma_j \sim N(0, \Gamma),
\]

(11)

where, \(y_{ij}\) represent the response for person \(i\) on item \(j\), \(i = 1, \ldots, I\). Each person is assumed to respond to an idiosyncratic set of \(j \in J_i\) items. This yields an unbalanced data set with a total of \(\sum_{i=1}^I J_i = N\) observations. Such a model arises, for instance, in recommendation systems where users rate different items (products). The vector \(x_{ij}\) includes covariates that characterize the persons and the items, \(z_j\) consists of item-specific covariates and \(w_i\) contains person-specific covariates, such as demographics. The vectors \(\lambda_i\)
and $\gamma_j$ include the person-specific coefficients and item-specific coefficients, respectively. In Appendix A we derive the variational distributions for this model.

To assess the speed and accuracy of the mean-field variational Bayesian (MFVB) approach, we now compare it to Gibbs sampling on simulated data sets of varying sizes. To ensure a fair comparison, we code both MFVB and Gibbs sampling in Mathematica 10 and compile the programs to C, using the built-in just-in-time compilation facilities available in Mathematica. The compilation results in highly efficient code, in particular for Gibbs sampling, which requires looping over full conditional draws. This is a typical computational bottleneck for interpreted languages, which often rely on vectorization for speed. However, vectorization is not an option in MCMC programs, and thus compilation offers significant speed benefits. The codes are run on a Mac Pro computer with 3GHz 8-Core Intel Xeon E5 processor and 32G of RAM.

Table 1 shows the sizes of the different data sets and the time needed for convergence in MFVB, using a tolerance of $10^{-4}$ for relative change in ELBO. The last column gives the time in seconds for the Gibbs sampling algorithm to finish 5000 iterations. Typically, MCMC methods require a much greater number of draws, especially for models involving multiple sources of heterogeneity. Thus our use of only 5000 iterations can be interpreted as a conservative estimate of the total number of of MCMC iterations required for convergence in such models. One can see from the table that MFVB requires very few iterations for convergence. It is also clearly apparent that the MFVB approach is considerably faster than MCMC and results in a substantial reduction in computational time. The last column of Table 1 reports the ratio of the time required for Gibbs sampling to that of MFVB. As the MFVB approach requires fewer iterations for larger data sets, we see that this ratio increases with data set size. It is clear from this that MFVB scales much better than MCMC for larger data sets.

<table>
<thead>
<tr>
<th>Persons $I$</th>
<th>Products $J$</th>
<th>Observations $I \times J$</th>
<th>MFVB (Tol = $10^{-4}$)</th>
<th>Gibbs Sampling (5000 iter.)</th>
<th>Speed Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td># Iter.</td>
<td>Time (sec.)</td>
<td># Iter.</td>
</tr>
<tr>
<td>300</td>
<td>50</td>
<td>15,000</td>
<td>7</td>
<td>0.26</td>
<td>136.56</td>
</tr>
<tr>
<td>3,000</td>
<td>50</td>
<td>150,000</td>
<td>6</td>
<td>2.05</td>
<td>1,338.84</td>
</tr>
<tr>
<td>3,000</td>
<td>500</td>
<td>1,500,000</td>
<td>3</td>
<td>6.64</td>
<td>13,642.13</td>
</tr>
<tr>
<td>30,000</td>
<td>500</td>
<td>15,000,000</td>
<td>3</td>
<td>114.44</td>
<td>593,138.32</td>
</tr>
</tbody>
</table>

To assess the accuracy with which the true parameter estimates are recovered by MFVB when compared
to MCMC, we simulate 10 different data sets and compute the root mean squared errors (RMSE) between the estimated and the true parameters. In each data set, we choose $I = 3000$, $J = 50$ and the total number of observations as 150,000. Across the 10 simulated data sets, the average RMSE across all parameters and its standard deviation are 0.338 and 0.006, respectively for MFVB estimation, when compared to 0.338 and 0.005, respectively for Gibbs sampling. This shows that the MFVB method produces parameter estimates that are as accurate as Gibbs sampling. Additional details regarding these simulations and the resulting estimates are available from the authors upon request. We now describe the fixed-form variational Bayes approach.

### 3.2 Fixed-form Variational Bayes (FFVB)

The fixed-form variational approximation (Honkela et al. 2010; Wang and Blei 2013; Knowles and Minka 2011; Salimans and Knowles 2013) can be used for estimating non-conjugate models. The FFVB method is analogous to the Metropolis-Hastings algorithm within the umbrella of MCMC in its applicability to a wide variety of non-conjugate models.

Recall that in the mean-field approach, the full conditional distributions dictate the functional form of the variational distributions. These full conditional distributions, however, are not available in closed-form in non-conjugate settings. In the FFVB approach, therefore, the variational distribution is assumed to belong to a particular family of distributions. This restriction implies that the variational distribution has a fixed functional form. For instance, when restricted to the exponential family (Wedel and Kamkura 2001), the variational distribution can be written as

$$q(\theta|\eta) = \nu(\theta) \exp \left( S(\theta)\eta - Z(\eta) \right),$$  \hspace{1cm} (12)

where, $\eta$ is the vector of natural parameters for the family, $S(\theta)$ contains the sufficient statistics of $\theta$, $Z(\eta)$ ensures normalization, and $\nu(\theta)$ is the base measure. The goal is to find $\eta$ that minimizes the KL divergence in (2), i.e.,

$$\eta^* = \arg \min_{\eta} \mathbb{E}_q \left[ \log q(\theta) - \log p(\theta|y) \right].$$  \hspace{1cm} (13)

A number of different approaches have been used to implement fixed-form variational Bayes. Wang and Blei (2013) suggest Laplace variational inference which is based on the Laplace approximation of the posterior. Knowles and Minka (2011) use non-conjugate variational message passing with the delta method (see also Bickel and Doksum 2007; Braun and McAuliffe 2010). Salimans and Knowles (2013) propose
stochastic linear regression, which we adopt in current paper for fixed-form VB, given its generality and accuracy.

Salimans and Knowles (2013) and Nott et al. (2013) use the properties of the exponential family to show that the optimization in (13) leads to a fixed point update for the variational parameter,

$$\eta = \text{Cov}_{q(\theta)}[S(\theta)^{-1}] \text{Cov}_{q(\theta)}[S(\theta), \log p(y, \theta)],$$  \hspace{1cm} (14)

where, Cov$_{q(\theta)}$ denotes the covariance with respect to the variational distribution. Instead of approximating Cov$_{q(\theta)}[S(\theta)^{-1}]$ and Cov$_{q(\theta)}[S(\theta), \log p(y, \theta)]$ directly, Salimans and Knowles (2013) iteratively evaluate these terms using weighted Monte Carlo with random samples of $\hat{\theta}$ generated from the latest variational approximation $q(\theta|\eta)$. When the multivariate normal is used to approximate the posterior, i.e., $q(\theta|\eta) = \mathcal{N}(\mu_{q(\theta)}, \Sigma_{q(\theta)})$, Minka (2001) and Opper and Archambeau (2009) show that (14) implies

$$\Sigma^{-1}_{q(\theta)} = -E_q[\partial^2 \log p(y, \theta)/\partial \theta^2] \quad \text{and} \quad \mu_{q(\theta)} = E_q[\theta] + \Sigma_{q(\theta)} E_q[\partial \log p(y, \theta)/\partial \theta],$$  \hspace{1cm} (15)

where $\partial/\partial \theta$ and $\partial^2/\partial \theta^2$ denote the gradient vector and Hessian matrix of $\log p(y, \theta)$, respectively. As in the general case, one can use weighted monte carlo to stochastically approximate the quantities, $H = -E_q[\partial^2 \log p(y, \theta)/\partial \theta^2]$, $g = E_q[\partial \log p(y, \theta)/\partial \theta]$, and $m = E_q[\theta]$. Algorithm 2 details the generic FFVB procedure underlying stochastic linear regression. For more details about the algorithm and the theory behind it, we refer the interested readers to Salimans and Knowles (2013) and Salimans (2014).

As in non-conjugate settings, an analytical expression for ELBO may not be available, we assess convergence based on the relative change in the Euclidean norm of the parameter estimates, compared to a desired tolerance level.

### 3.2.1 A Logit Model

To illustrate the FFVB method, and to assess its performance relative to MCMC methods, we conduct a simulation study using a homogeneous conditional logit model. Suppose data consists of $I$ observations involving a choice among $J$ alternatives. The logit choice probability is given by

$$p(y_{ij} = 1) = \frac{\exp(x'_{ij}\beta)}{\sum_{k=1}^{J} \exp(x'_{ik}\beta)},$$  \hspace{1cm} (16)

where $i = 1, \ldots, I$ and $j = 1, \ldots, J$. 

Algorithm 2. FFVB for \( q(\theta) = N(\mu_{q(\theta)}, \Sigma_{q(\theta)}) \)

1. Initialize \( \mu_{q(\theta)}, \Sigma_{q(\theta)}, H, g \) and \( m \).
2. Initialize \( H = 0, g = 0 \) and \( m = 0 \).
3. Set the total number of iterations \( M \) and step size \( \omega = 1/\sqrt{M} \).
4. At each iteration \( n = 1, \ldots, M \):
   (a) Generate a draw \( \hat{\theta} \) from \( N(\mu_{q(\theta)}, \Sigma_{q(\theta)}) \).
   (b) Calculate the gradient \( \hat{g} \) and the Hessian \( \hat{H} \) of \( \log p(y, \theta) \) at \( \hat{\theta} \).
   (c) Set \( g = (1 - \omega)g + \omega \hat{g}, H = (1 - \omega)H - \omega \hat{H} \) and \( m = (1 - \omega)m + \omega \hat{\theta} \).
   (d) Update \( \Sigma_{q(\theta)} = H^{-1} \) and \( \mu_{q(\theta)} = \Sigma_{q(\theta)}g + m \).
   (e) If \( n > M/2 \), then \( \bar{g} = g + \frac{2}{M} \hat{g}, \bar{H} = H - \frac{2}{M} \hat{H} \) and \( \bar{m} = m + \frac{2}{M} \hat{\theta} \).
5. Set \( \Sigma_{q(\theta)} = \bar{H}^{-1} \) and \( \mu_{q(\theta)} = \Sigma_{q(\theta)}\bar{g} + \bar{m} \).

We specify a normal prior \( N(\mu_\beta, \Sigma_\beta) \) for the coefficient vector \( \beta \). We then assume that the variational distribution is multivariate normal. Let \( \bar{x}_i = \sum_{j=1}^{J} p(y_{ij} = 1) \cdot x_{ij} \). The gradient and Hessian of the logarithm of the unnormalized posterior can be derived as

\[
\frac{\partial \log p(y, \beta)}{\partial \beta} = \sum_{i=1}^{I} \sum_{j=1}^{J} y_{ij} (x_{ij} - \bar{x}_i) - \Sigma_\beta^{-1} (\beta - \mu_\beta) \tag{17}
\]

\[
\frac{\partial^2 \log p(y, \beta)}{\partial \beta^2} = - \sum_{i=1}^{I} \sum_{j=1}^{J} p(y_{ij} = 1) \cdot (x_{ij} - \bar{x}_i) (x_{ij} - \bar{x}_i)' - \Sigma_\beta^{-1} \tag{18}
\]

By plugging in these derivatives in Algorithm 2 and setting \( \theta = \beta \), we can perform FFVB for the conditional logit model to estimate \( \beta \).

We simulate multiple data sets with varying sizes according to (16), and compare FFVB and MCMC in Table 2. In each data set, we use a choice set of five alternatives on each choice occasion. One can see from the table that the variational approach is significantly faster (about 60 times) when compared with MCMC, and is therefore more suitable for large choice data sets.

To assess accuracy, we simulate 10 different choice data sets using (16) and compare the estimation results obtained from FFVB and the Metropolis-Hastings method. In each data set, we use 100,000 choice occasions. As the true parameter values that generate the data sets are known, we compute RMSEs be-
Table 2: Compare FFVB with MCMC for the Logit Model

<table>
<thead>
<tr>
<th># Obs.</th>
<th># Iter.</th>
<th>Time (sec.)</th>
<th>Time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10,000</td>
<td>66</td>
<td>0.68</td>
<td>37.45</td>
</tr>
<tr>
<td>100,000</td>
<td>66</td>
<td>5.62</td>
<td>368.10</td>
</tr>
<tr>
<td>1,000,000</td>
<td>66</td>
<td>63.44</td>
<td>3,767.27</td>
</tr>
<tr>
<td>10,000,000</td>
<td>67</td>
<td>716.65</td>
<td>40,773.90</td>
</tr>
</tbody>
</table>

The derivation of the gradient and Hessian of the log-likelihood is model-specific. For the logit model, the derivation is straightforward. However, for other models, these derivations could become tedious if the log-likelihood is complex. In such cases, however, one could use the algorithmic or automatic differentiation, such as MAD within MATLAB (Forth 2006), Theano within Python (Bastien et al. 2012) and the symbolic differentiation functions within Mathematica (Wolfram Research, Inc. 2014), to compute these quantities without the need for explicit derivation. Next, we show an application of FFVB where we rely on Mathematica’s symbolic differentiation capabilities to efficiently and exactly calculate the gradients and Hessians.

3.2.2 A Heckman Selection Model

Heckman selection models (or Type-2 Tobit models) offer a popular mechanism to account for selectivity in econometric and marketing situations (Heckman 1979; Nelson 1977; Wooldridge 2002; Ying, Feinberg and Wedel 2006). Selectivity arises from the presence of latent variables that censor the outcome of interest. For instance, when studying purchase quantities of a certain brand, the quantity that is bought is observed only if the buyer decides to buy. More formally, denote $y_2^*$ as the outcome of interest, e.g., purchase quantity, and $y_1^*$ be the censoring latent variable, e.g., utility of buying. Purchase quantities are observed only if the utility of buying exceeds a threshold, i.e., $y_1^* > 0$, indicating that the buyer decided to make a purchase. Formally,
the participation decision of whether to buy is given by

\[ y_1 = \begin{cases} 
1, & \text{if } y_1^* \geq 0 \\
0, & \text{if } y_1^* < 0,
\end{cases} \]  \hspace{1cm} (19)

and the censored quantity is written as

\[ y_2 = \begin{cases} 
y_2^*, & \text{if } y_1^* \geq 0 \\
-, & \text{if } y_1^* < 0.
\end{cases} \]  \hspace{1cm} (20)

In above, \( y_2 \) need not take on any meaningful value when \( y_1 < 0 \). Often, the goal is to understand the drivers of the censored variable (e.g., purchase quantity). It is common to use a linear specification for the latent variables in terms of covariates \( x_1 \) and \( x_2 \) as follows

\[
y_1^* = x_1 \beta_1 + \epsilon_1,
\]

\[
y_2^* = x_2 \beta_2 + \epsilon_2.
\]  \hspace{1cm} (21)

Assuming \( \epsilon_1 \) and \( \epsilon_2 \) are joint normally distributed,

\[
\begin{pmatrix} \epsilon_1 \\ \epsilon_2 \end{pmatrix} \sim \mathcal{N} \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix} \right).
\]

The log-likelihood of this model can then be written as

\[
\log p(y_1, y_2 | \beta_1, \beta_2) = \sum_{i=1}^I \left( (1 - y_{1i}) \log p(y_{1i}^* < 0) + y_{1i} \log p(y_{2i} | y_{1i}^* \geq 0) + y_{1i} \log p(y_{1i}^* \geq 0) \right). \hspace{1cm} (22)
\]

The derivation for the gradient and Hessian with respect to model parameters becomes tedious in such a model. We therefore use the symbolic differentiation capabilities within Mathematica to compute the gradient and Hessian of the log-likelihood. Mathematica’s meta-programming capabilities also allow us to compile these expressions into C functions that allow fast computation.

We apply the FFVB schedule in Algorithm 2 to estimate the Heckman selection model. To compare the performance between FFVB and MCMC, we simulate multiple data sets with varying sizes according to (19), (20) and (21). Table 3 shows the results on estimation speed. One can see from the table that the variational approach is substantially faster (about 5 to 8 times) when compared with MCMC. It is important to note that our MCMC algorithm relies on the adaptive Metropolis-Hastings method. Data augmentation solutions that rely on the Gibbs sampler (Li 1998; McCulloch et al. 2000) can alternatively be used. However,
these are likely to be computationally inefficient as they augment the latent and partially latent variables on every observation. Given the large number of observations in a Big Data situation, such a strategy can be computationally prohibitive.

To test the accuracy of the FFVB estimates, we simulate 10 different data sets, each with 100,000 observations. We then compared the parameter estimates obtained from FFVB and the Metropolis-Hastings method. RMSEs are calculated between the estimates and the true model parameters. Across the 10 runs, the average and standard deviation of RMSEs are 0.009 and 0.005, respectively, for MFVB estimation, as opposed to 0.008 and 0.003, for Metropolis-Hastings. Therefore, FFVB offers the advantage of faster estimation, compared with MCMC, and also recovers the parameters accurately.

Table 3: Compare FFVB with MCMC for the Heckman Selection Model

<table>
<thead>
<tr>
<th></th>
<th>FFVB (Tol = 10^{-6})</th>
<th>Metropolis-Hastings (20000 iter.)</th>
</tr>
</thead>
<tbody>
<tr>
<td># Obs.</td>
<td># Iter.</td>
<td>Time (sec.)</td>
</tr>
<tr>
<td>100,000</td>
<td>684</td>
<td>55.14</td>
</tr>
<tr>
<td>500,000</td>
<td>808</td>
<td>415.46</td>
</tr>
<tr>
<td>1,000,000</td>
<td>955</td>
<td>1019.34</td>
</tr>
</tbody>
</table>

4 Hybrid Variational Bayes

In marketing, we often face complex panel data situations that require hierarchical models with both conjugate and non-conjugate components. In such models, we can use a hybrid approach that combines both mean-field and fixed-form VB. Such an approach is similar to what is popularly known as the Metropolis within Gibbs strategy in the MCMC tradition. We discuss the hybrid approach in the context of a general hierarchical model with data from $I$ individuals, $\{y_i\}_{i=1}^I$. We assume a model structure that includes individual-specific parameters $\lambda_i$, and two sets of hyper-parameters, $\theta_1$ and $\theta_2$. The dependency structure of the model is given by the joint distribution,

$$p(\theta_1, \theta_2; \{\lambda_i\}, \{y_i\}) = \prod_{i=1}^I p(y_i|\lambda_i, \theta_1)p(\lambda_i|\theta_2)p(\theta_2)p(\theta_1),$$

and the directed acyclic graph of the model is represented in Figure 1.
The resulting augmented posterior distribution is given by
\[
p(\theta_1, \theta_2, \{\lambda_i\}, | \{y_i\}) = \frac{p(\{y_i\} | \theta_1, \theta_2, \{\lambda_i\})p(\theta_1, \theta_2, \{\lambda_i\})}{\int p(\theta_1, \theta_2, \{\lambda_i\}, \{y_i\}) d\theta_1 d\theta_2 d\lambda_1 \ldots d\lambda_I}.
\] (24)

Non-conjugacy in such a model can arise from two sources. First, the likelihood \( \prod_{i=1}^I p(y_i | \lambda_i, \theta_1) \) may not be conjugate with the population distribution \( p(\lambda_i | \theta_2) \) and therefore, the full conditionals for the individual-level coefficients, \( \lambda_i, \forall i \), may not be available in closed-form. Second, we assume that the prior \( p(\theta_1) \) is not conjugate to the likelihood. Regarding the population distribution, we assume that it belongs to the exponential family (the multivariate normal being the most commonly used choice). Due to this restriction to the exponential family, the priors \( p(\theta_2) \) can be chosen to be conjugate or semi-conjugate to the population distribution.

Given the above fairly general model structure, the variational distribution can be specified in a factorized form for both the conjugate and non-conjugate components as follows
\[
q(\{\lambda_i\}, \theta_1, \theta_2) = \prod_{i} q(\lambda_i)q(\theta_1)q(\theta_2),
\] (25)

where, the non-conjugate factors \( q(\lambda_i) \) and \( q(\theta_1) \) have fixed-forms (such as multivariate normal), and the conjugate factor \( q(\theta_2) \) can be optimally deduced from the known full conditional for \( \theta_2 \). The resulting estimation process involves using stochastic linear regression (Algorithm 2) for \( \{\lambda_i\}_{\forall i}, \theta_1 \), which serves
as an inner loop within an outer loop that updates the conjugate parameters $\theta_2$ using coordinate ascent. We now illustrate the hybrid framework within the context of a hierarchical ordinal logit model.

### 4.1 Hierarchical Ordinal Logit Model

Ordinal logit models are used for modeling ordinal responses (Train 2009). Marketing examples include modeling quantities (Balasubramanian et al. 1988) or responses on a rating scale, as in movie ratings within a movie recommendation system (Ying, Feinberg and Wedel 2006). In the latter application, it is common to model the observed responses using a latent utility, $u_{ij}$, that represents user $i$’s preference for the $j$th product and a set of thresholds. It is further assumed that the observed rating changes whenever the underlying utility crosses a threshold. For example, in the movie recommendation context, user $i$’s ratings can be represented as follows

$$
\begin{align*}
  r_{ij} = \begin{cases} 
  1, & u_{ij} < 0 \\
  2, & 0 < u_{ij} < \tau_1 \\
  3, & \tau_1 < u_{ij} < \tau_2 \\
  \vdots \ & \vdots \\
  R, & u_{ij} > \tau_L,
  \end{cases}
\end{align*}
$$

where $\{\tau_1, \ldots, \tau_L\}$ are the threshold cutoffs for the $R = L + 2$ rating categories. Assuming a random utility framework, the utility of user $i$ on item $j$ can be expressed as a function of both observed variables and unobserved components, $u_{ij} = x'_{ij} \lambda_i + \epsilon_{ij}$. When the distribution of the unobservable $\epsilon_{ij}$ is logistic, i.e., $F(\epsilon_{ij}) = \exp(\epsilon_{ij})/(1 + \exp(\epsilon_{ij}))$, we obtain an ordinal logit model. The probability of observing a rating $l$ can be written as

$$
p(\tau_{l-2} < u_{ij} < \tau_{l-1}) = p(\epsilon_{ij} < \tau_{l-1} - x'_{ij} \lambda_i) - p(\epsilon_{ij} < \tau_{l-2} - x'_{ij} \lambda_i)
\leq \frac{\exp(\tau_{l-1} - x'_{ij} \lambda_i)}{1 + \exp(\tau_{l-1} - x'_{ij} \lambda_i)} - \frac{\exp(\tau_{l-2} - x'_{ij} \lambda_i)}{1 + \exp(\tau_{l-2} - x'_{ij} \lambda_i)}.
\leq (26)
$$

In this model, we assume that the individual specific parameters $\lambda_i$’s come from a multivariate normal population distribution, $\lambda_i \sim \text{N}(\beta, \Lambda)$. We use typical semi-conjugate hyperpriors for the hyperparameters, i.e., a multivariate normal prior $\beta \sim \text{N}(\mu_\beta, \Sigma_\beta)$ and an Inverse Wishart prior, $\Lambda \sim \text{IW}(\rho_\Lambda, R_\Lambda)$. To ensure that the cutoffs are properly ordered we use a reparametrisation such that $\tau_l = \sum_{k=1}^l \exp(\kappa_k)$. We then use a multivariate normal prior for the transformed parameter vector, i.e., $\kappa \sim \text{N}(\mu_\kappa, \Sigma_\kappa)$. 

17
The hierarchical setting results in a model setup that is a specific instantiation of the general hybrid framework outlined earlier. The hyperparameters for the normal population distribution have closed-form full conditionals thus enabling MFVB for the population mean $\beta$, and the population covariance $\Lambda$. The individual-level parameters $\lambda_i$, $\forall i$, and the vector of reparametrised cutoffs $\kappa$ require FFVB. We therefore assume a factorized form for the variational approximation to the true posterior,

$$ q(\beta, \Lambda, \kappa, \{\lambda_i\}) = q(\beta)q(\Lambda)q(\kappa)\prod_{i=1}^{M}q(\lambda_i) $$

Conditional on $\lambda_i$’s, we use the MFVB setup to deduce the closed-form variational distributions $q(\beta) = N(\mu_{q(\beta)}, \Sigma_{q(\beta)})$ and $q(\Lambda) = IW(\rho_{q(\Lambda)}, R_{q(\Lambda)})$. For the non-conjugate parameters, we use a multivariate normal approximation for the variational distributions, i.e., $q(\lambda_i) = N(\mu_{q(\lambda_i)}, \Sigma_{q(\lambda_i)})$ and $q(\kappa) = N(\mu_{q(\kappa)}, \Sigma_{q(\kappa)})$. Following the above outlined hybrid framework, we embed the FFVB updating as an inner loop within the outer MFVB iterations to estimate the hierarchical ordinal logit model. Algorithm 3 provides the detailed procedure.

We have seen that variational Bayesian methods require significantly less time when compared to MCMC methods for parameter estimation. This relative advantage of VB can be further enhanced using stochastic variational inference and parallelization. We next discuss these approaches and then apply them within the ordinal logit context.

## 5 Speeding Up Variational Bayes

When fitting a complex model with many latent variables (such as the hierarchical ordinal logit shown above) to a big data set, the hybrid variational inference procedure requires significant computation because of need for the gradient and Hessian for every individual. Recent research has explored algorithmic and computational strategies to speed up this computation via stochastic variational inference (Hoffman et al. 2013; Tan 2014). Moreover, given the widespread availability of computers with multiple cores and the ease of access to cloud computing, parallelization can also be readily used via a “divide and recombine” strategy (Huang and Gelman 2005; Nott et al. 2013). We apply these strategies, singly and in tandem to the VB estimation of the hierarchical ordinal logit model, and show in Section 6 that an integration of these two methods can provide significant improvements to the speed and scalability of variational Bayesian inference. Furthermore, as we will discuss later, parallelization can also be used to address streaming data involving high velocity.
Algorithm 3. Hybrid VB for the ordinal Logit Model

1. Initialize $\mu_{q(\theta)}$, $\Sigma_{q(\theta)}$, $\mu_{q(\kappa)}$, $\Sigma_{q(\kappa)}$, $\{\mu_{q(\lambda_i)}\}$ for all $i$, $\{\Sigma_{q(\lambda_i)}\}$ for all $i$, $\rho_{q(\Lambda)}$, and $R_{q(\Lambda)}$.

2. Set the number of FFVB inner iterations $M$, and step size $\omega$.

3. Update for $\lambda_i$, $\forall i$, as follows
   (1) Initialize $H_{\lambda_i} = \Sigma_{q(\lambda_i)}^{-1}$, $g_{\lambda_i} = \mu_{q(\lambda_i)}$, $m_{\lambda_i} = 0$.
   (2) Initialize $\tilde{H}_{\lambda_i} = 0$, $\tilde{g}_{\lambda_i} = 0$, $\tilde{m}_{\lambda_i} = 0$.
   (3) At each iteration $n = 1, \ldots, M$:
       (a) Generate a draw $\hat{\lambda}_i$ from $N(\mu_{q(\lambda_i)}, \Sigma_{q(\lambda_i)})$.
       (b) Calculate the gradient $\hat{g}_{\lambda_i}$ and Hessian $H_{\lambda_i}$ of $\log p(y_i; \{\lambda_i\}_{\forall i}, \kappa)$ at $\hat{\lambda}_i$.
       (c) Set $g_{\lambda_i} = (1 - \omega)g_{\lambda_i} + \omega \hat{g}_{\lambda_i}$, $H_{\lambda_i} = (1 - \omega)H_{\lambda_i} - \omega \hat{H}_{\lambda_i}$, and $m_{\lambda_i} = (1 - \omega)m_{\lambda_i} + \omega \hat{\lambda}_i$.
       (d) Update $\Sigma_{q(\lambda_i)} = H_{\lambda_i}^{-1}$ and $\mu_{q(\lambda_i)} = \Sigma_{q(\lambda_i)}g_{\lambda_i} + m_{\lambda_i}$.
       (e) If $n > M/2$, then $\tilde{g}_{\lambda_i} = g_{\lambda_i} + \frac{2}{M} \hat{g}_{\lambda_i}$, $\tilde{H}_{\lambda_i} = H_{\lambda_i} - \frac{2}{M} \hat{H}_{\lambda_i}$, and $\tilde{m}_{\lambda_i} = m_{\lambda_i} + \frac{2}{M} \hat{\lambda}_i$.
   (4) Set $\Sigma_{q(\lambda_i)} = \tilde{H}_{\lambda_i}^{-1}$ and $\mu_{q(\lambda_i)} = \Sigma_{q(\lambda_i)}\tilde{g}_{\lambda_i} + \tilde{m}_{\lambda_i}$.

4. Update for $\kappa_i$, $\forall i$, as follows
   (1) Initialize $H_{\kappa} = \Sigma_{q(\kappa)}^{-1}$, $g_{\kappa} = \mu_{q(\kappa)}$, $m_{\kappa} = 0$.
   (2) Initialize $\tilde{H}_{\kappa} = 0$, $\tilde{g}_{\kappa} = 0$, $\tilde{m}_{\kappa} = 0$.
   (3) At each iteration $n = 1, \ldots, M$:
       (a) Generate a draw $\hat{\kappa}$ from $N(\mu_{q(\kappa)}, \Sigma_{q(\kappa)})$.
       (b) Calculate the gradient $\hat{g}_{\kappa}$ and Hessian $H_{\kappa}$ of $\log p(y_i; \{\lambda_i\}_{\forall i}, \kappa)$ at $\hat{\kappa}$.
       (c) Set $g_{\kappa} = (1 - \omega)g_{\kappa} + \omega \hat{g}_{\kappa}$, $H_{\kappa} = (1 - \omega)H_{\kappa} - \omega \hat{H}_{\kappa}$, and $m_{\kappa} = (1 - \omega)m_{\kappa} + \omega \hat{\kappa}$.
       (d) Update $\Sigma_{q(\kappa)} = H_{\kappa}^{-1}$ and $\mu_{q(\kappa)} = \Sigma_{q(\kappa)}g_{\kappa} + m_{\kappa}$.
       (e) If $n > M/2$, then $\tilde{g}_{\kappa} = g_{\kappa} + \frac{2}{M} \hat{g}_{\kappa}$, $\tilde{H}_{\kappa} = H_{\kappa} - \frac{2}{M} \hat{H}_{\kappa}$, and $\tilde{m}_{\kappa} = m_{\kappa} + \frac{2}{M} \hat{\kappa}$.
   (4) Set $\Sigma_{q(\kappa)} = \tilde{H}_{\kappa}^{-1}$ and $\mu_{q(\kappa)} = \Sigma_{q(\kappa)}\tilde{g}_{\kappa} + \tilde{m}_{\kappa}$.

5. Update $\Sigma_{q(\beta)} = (\Sigma_{q(\beta)}^{-1} + I\rho_{q(\Lambda)}R_{q(\Lambda)}^{-1})^{-1}$ and $\mu_{q(\beta)} = \Sigma_{q(\beta)}(\Sigma_{q(\beta)}^{-1}\mu_{q(\beta)} + \rho_{q(\Lambda)}R_{q(\Lambda)}^{-1}\sum_{i=1}^{I} \mu_{q(\lambda_i)})$.

6. Update $\rho_{q(\Lambda)} = \rho_{\Lambda} + I$ and $R_{q(\Lambda)} = R_{\Lambda} + I\Sigma_{q(\beta)} + \sum_{i=1}^{I} ((\mu_{q(\lambda_i)} - \mu_{q(\beta)})(\mu_{q(\lambda_i)} - \mu_{q(\beta)})') + \Sigma_{q(\lambda_i)}$.

7. Repeat Steps 3-6 until convergence.
5.1 Parallelization

In the parallelization strategy, the original data is partitioned into independent subsets and variational Bayesian inference is performed separately on each subset of the data. The resulting variational Bayesian distributions on the subsets are then combined appropriately to yield overall estimates of model parameters (Huang and Gelman 2005; Nott et al. 2013).

We can partition the entire data set $y$ into $K$ subsets, $\{y_1, \ldots, y_K\}$ that are conditionally independent, given the model parameter $\theta$. When dealing with panel data, it is important to ensure that all observations for a given individual are kept together as part of the same subset. The posterior distribution after the “division” becomes

$$p(\theta|y) \propto p(\theta)p(y|\theta)$$

$$= p(\theta) \prod_{k=1}^{K} p(y_k|\theta)$$

$$= p(\theta)^{1-K} \prod_{k=1}^{K} \{p(\theta)p(y_k|\theta)\}$$

$$\propto p(\theta)^{1-K} \prod_{k=1}^{K} p(\theta|y_k). \quad (27)$$

That is, the posterior can be obtained as a combination of the separate posteriors from each data subset. This allows us to compute the variational approximations $q(\theta|\eta_k)$ of each subset posterior $p(\theta|y_k)$, $k = 1, \ldots, K$, independently, in parallel, using the different subsets on different processing units (kernels) simultaneously. Such parallelization can result in considerable savings in computation time because each variational approximation happens on a much smaller subset of the original data. Moreover, as the partitioning into the subsets happens once in the beginning and computation proceeds independently on each processing unit without communication overhead between the processing units, parallelization leads to significant improvements in speed. After each sub-computation has converged, the subset variational approximations are recombined to yield the overall approximation to the posterior of interest

$$p(\theta|y) \propto p(\theta)^{1-K} \prod_{k=1}^{K} q(\theta|\eta_k). \quad (28)$$

The parallelization strategy is particularly useful for variational distributions from the exponential family, which we already described in (12). Suppose $\eta_0$ is the natural parameter of the prior $p(\theta)$, and $\eta_k$ of
the variational distribution on the $k$th subset of the data, the recombination in (28) implies that the natural parameter of $p(\theta | y)$ can be computed as

$$\eta = \sum_{k=1}^{K} \eta_k - (K - 1) \eta_0.$$  \hspace{1cm} (29)

We can translate the above general result to the context of an ordinal logit model. Note that the variational distribution for $\beta$ is a normal. Suppose that we separately estimate $q(\beta)^{(k)} = N(\mu_{q(\beta)}^{(k)}, \Sigma_{q(\beta)}^{(k)})$ for all the data subsets, where the superscript indicates the $k$th data subset, then the variational parameters in $q(\beta) = N(\mu_{q(\beta)}, \Sigma_{q(\beta)})$ for the whole data set are

$$\Sigma_{q(\beta)}^{-1} = \frac{1}{K} \sum_{k=1}^{K} \left( \Sigma_{q(\beta)}^{(k)} \right)^{-1} - (K - 1) \Sigma_{q(\beta)}^{-1} \quad \text{and} \quad \mu_{q(\beta)} = \frac{1}{K} \sum_{k=1}^{K} \left( \Sigma_{q(\beta)}^{(k)} \right)^{-1} \mu_{q(\beta)}^{(k)} - (K - 1) \mu_{q(\beta)}.$$

The variational distributions for the cutoff vector $\kappa$ can also be combined in a similar fashion.

As for $q(\Lambda) = IW(\rho_{q(\Lambda)}, R_{q(\Lambda)})$, once we obtain the separate variational estimates, $IW(\rho_{q(\Lambda)}^{(k)}, R_{q(\Lambda)}^{(k)})$, from the independent runs, the combined variational parameters can be computed as

$$\rho_{q(\Lambda)} = \sum_{k=1}^{K} \rho_{q(\Lambda)}^{(k)} - (K - 1) \rho_{\Lambda} \quad \text{and} \quad R_{q(\Lambda)} = \sum_{k=1}^{K} R_{q(\Lambda)}^{(k)} - (K - 1) R_{\Lambda}.$$  \hspace{1cm} (31)

The above discusses how parallelization can be used. We now describe how a given VB computation can be further speeded up using stochastic variation inference (Hoffman et al. 2013; Tan and Nott 2014).

### 5.2 Stochastic Optimization with Adaptive Mini-batch

To understand how stochastic variational inference works, it is important to distinguish between global and local model parameters in hierarchical models. Global parameters are invariant across individuals, whereas local parameters are individual-specific. For instance, in the hierarchical ordinal logit model, the population parameters $\beta$, $\Lambda$ and the thresholds $\kappa$ are global parameters, and the random coefficients $\lambda_i$’s are local parameters as they vary across individuals. A closer look at Algorithm 3 shows that a single update of the global parameters requires us to update all the local parameters. This is computationally wasteful because in the early VB iterations, these local parameter updates are based upon the unconverged estimates of global parameters. Moreover, the extent of the inefficiency worsens when we have a large number of individuals in
the data. Stochastic variational inference aims to address this computational bottleneck by eliminating the need to make a complete pass through the entire data set as it uses only mini-batches of individuals (rather than all individuals) in each iteration of the VB.

Hoffman et al. (2013) develops a stochastic variational Bayesian approach that relies on stochastic gradient descent methods (Robins and Monro 1951). In this approach, one uses a randomly selected subset of the whole data set – a mini-batch, \( B \), for every iteration of the VB. Given the current values of the global parameters, the local parameters are then updated only for the individuals within the mini-batch, \( B \). Finally, the global variational parameters are updated using only the updated local parameters within the mini-batch. Note that the mini-batch \( B \) is of the same size across the iterations, but it contains a randomly selected sample of individuals in each iteration. Such random selection ensures that, although a given iteration works from the mini-batch, all the data is used across the iterations. This use of mini-batches is different from the parallelization strategy where the subsets of a given data set remain fixed in size and content across iterations.

Spall (2003) show that, under certain regularity conditions, such a stochastic optimization process can probabilistically converge to an optimum. Hoffman et al. (2013) apply this method to topics models and demonstrate that the use of mini-batches can result in substantial computational savings. Additional computational savings can be obtained by adaptively changing the step size (Ranganath et al. 2013) or the mini-batch size (Tan 2014) across the VB iterations. In the above, we provided a brief overview of stochastic optimization. The interested reader can understand further the technical details of these adaptation strategies from the above sources.

In this paper, we adapt the size of the mini-batch sample across iterations. Furthermore, we show how stochastic variational Bayesian methods can be used in combination with the parallelization strategy discussed above. We now showcase the hybrid variational approach with these computational strategies on simulated data from an ordinal logit model.

6 Ordinal Logit Model on Simulated Data

For the simulation, we generate a hierarchical data set of ordinal choices from \( I = 10,000 \) individuals and \( J = 20 \) occasions, i.e., with a total of 200,000 observations. On each occasion, the individual is assumed to select one out of five rating categories. In accounting for heterogeneity, we choose to specify
a full covariance matrix $\Lambda$ for the population distribution. Algorithm 3 and its variants to accommodate stochastic variational inference and parallelization are used on this simulated data. We use a convergence criterion of $10^{-6}$ for the relative change in parameter values. For the MCMC run, we use an algorithm that combines adaptive metropolis steps with Gibbs sampling steps. Given the nonlinearity of the model, we run the MCMC algorithm for 100,000 iterations. The computational time in MCMC scales linearly with the number of iterations.

Table 4 shows the results and compares the performance across different variational approaches. As we can see in the table, parallelization in combination with mini-batch sampling results in a dramatic reduction in computational time for the VB approaches, thereby offering a substantial speed advantage over MCMC. To assess accuracy, we hold out 25\% of the data of each individual for prediction and use the rest 75\% for model estimation. The hit rates of the VB approaches are similar or slightly higher when compared with MCMC. Therefore, VB estimation serves an accurate and much faster alternative to MCMC methods in current setting.

7 Real Data Application

In this section we apply the variational Bayesian method to the MovieLens data. MovieLens is a website that collects users’ film preferences and makes personalized recommendations of films for the users to watch. Our goal is to compare the variational Bayesian inference with MCMC on recovering preferences from the large database. A fast and accurate estimation of user preference is of great importance to every recommendation system, and our results show that the proposed variational Bayesian framework can satisfactorily achieve this goal.

The MovieLens data sets are collected and assembled by the GroupLens Research Project at the Uni-
The data contains ratings (from 1 to 5) provided by users for many different movies. Users differ in the number of films they rated, thereby yielding an unbalanced data set. The data consists of $N = 100,000$ ratings from $I = 943$ users on $J = 1682$ movies. Each user rated at least 20 movies. From the data we know for each movie its genre. There are 19 genres in total, and each film can be in several genres at once.

We apply the hierarchical ordinal logit model in (26) to the MovieLens data. We include movie genres in $x_{ij}$ and assign to it the person-specific coefficient $\lambda_i$. We specify prior distributions according to the discussion in Section 4.1. Table 5 compares VB with MCMC on estimation speed. It can be seen that, regardless of the model type, the variational approach offers the benefit of significant speedups when compared to MCMC. It is also apparent that parallelization of VB can substantially save computational time.
Table 5: Compare Variational Bayes with MCMC on MovieLens Rating Data

<table>
<thead>
<tr>
<th></th>
<th>Hybrid VB</th>
<th>Adaptive Mini-batch</th>
<th>Parallelization</th>
<th>Adaptive Mini-batch &amp; Parallelization</th>
<th>MCMC</th>
</tr>
</thead>
<tbody>
<tr>
<td># Iterations</td>
<td>386</td>
<td>412</td>
<td>61</td>
<td>97</td>
<td>100,000</td>
</tr>
<tr>
<td>Time (sec.)</td>
<td>2,960.96</td>
<td>2768.33</td>
<td>271.84</td>
<td>168.98</td>
<td>49,586.15</td>
</tr>
</tbody>
</table>

when handling massive marketing data sets. Figure 2 shows the convergence of the parameter vector norm, as well as the changing size of mini-batch across interactions. From the figure one can discern the benefit of adaptive mini-batch in that, before the mini-batch reaches the size of the full data set, the parameters have sufficiently progressed near the optima.³

8 Conclusion

The rapid growth in customer databases that characterizes the Big Data era has opened up opportunities to understand individual customers, and to customize offerings at much finer detail and at a faster pace than ever before. These insights on individual customer behavior could be crucial for the strategic and tactical success of firms in this era of Big Data (Brynjolfsson et al. 2011). Firms therefore need to estimate individual-level response sensitivities accurately and in a timely fashion, and also need to update their knowledge about customers as new data arrive. Traditionally, these issues could be addressed via Bayesian inference with MCMC methods (Rossi et al. 2005). However, as data sets become large and as models that support such data become increasingly complex, MCMC suffers from its computational expense.

Variational Bayesian inference offers a versatile solution to the problems that arise with big marketing data. As we have shown in this paper on several commonly used marketing models, VB methods can recover true parameters with the same accuracy as MCMC, but with a speed that could potentially be many times faster than MCMC. In terms of wall clock time, this means the time required to estimate or re-estimate hierarchical models that yield individual customer preferences would range from minutes to hours, instead of days to weeks – a crucial difference in the demand for rapid business intelligence that characterizes the current digital economy. We also show that recent innovations in parallelization and adaptive mini-batch can be used to further enhance the speed of VB. While we do not focus on data velocity and streaming data in this paper, it is apparent that parallelization techniques can enable managers to estimate parameters for

³There are a large number of parameter estimates for the MovieLens data. They are available upon request from the authors.
new customers or old customers with new observations, as a subset, without having to run the model for the entire population. This can yield considerable time savings and can address the issue of high data velocity.

In summary, the attractive features of VB for marketing are (1) the fast estimation speed compared to MCMC, with comparable estimation accuracy and without the necessity to throw away information via data-pruning or sampling, (2) the superior ability to scale to larger data sets, and (3) the flexibility of mean-field and fixed-form methods, analogous to Gibbs and Metropolis-Hasting, in handling a variety of conjugate and non-conjugate models that are widely used in marketing applications.

In this research we have illustrated the key advantages of VB for marketing models. We focus on the introduction of the VB framework for marketers. There are many potential avenues for future research and these can test the efficacy of VB methods for other model forms. For example, one can apply VB to even more complex situations such as those with multiple decisions, for handling dynamics, or for structural models. For streaming data, such as the browsing behavior and choice behavior of customers using Netflix, Amazon, or any e-commerce websites, one can assess how well VB performs, as a component of particle filtering (Smidl and Quinn 2008), or in tandem with parallelization and stochastic optimization, against the time-stamp, and formally evaluate its ability to deal with big data velocity. From a market research prospective, VB also has the potential to be useful in text mining, given its various successful applications within the latent Dirichlet allocation framework (Blei, Ng, and Jordan 2003), thus addressing the volume, velocity and variety – the 3Vs of big data, all at once.
References


Appendices

A Variational Distributions for Linear Models

The priors are

\[ p(\beta) = N(\mu_\beta, \Sigma_\beta), \quad p(\lambda_i) = N(0, \Lambda), \quad p(\Lambda) = IW(\rho_\Lambda, R_\Lambda) \]
\[ p(\gamma_j) = N(0, \Gamma), \quad p(\Gamma) = IW(\rho_\Gamma, R_\Gamma), \quad p(\sigma^2) = IG(a, b) \]

The full conditionals are

1) \( p(\beta | \text{rest}) = \mathcal{N}(\tilde{\mu}_\beta, \tilde{\Sigma}_\beta) \)

\[ \tilde{\Sigma}_\beta^{-1} = \Sigma_\beta^{-1} + \sigma^{-2} X'X \quad \text{and} \quad \tilde{\mu}_\beta = \tilde{\Sigma}_\beta (\Sigma_\beta^{-1} \mu_\beta + \sigma^{-2} (X'Y - \sum_{i=1}^I X_i Z_i \lambda_i - \sum_{j=1}^J X_j W_j \gamma_j)) \]

2) \( p(\lambda_i | \text{rest}) = \mathcal{N}(\tilde{\mu}_\lambda_i, \tilde{\Sigma}_\lambda_i) \)

\[ \tilde{\Sigma}_{\lambda_i}^{-1} = \Lambda_i^{-1} + \sigma^{-2} Z_i'Z_i \quad \text{and} \quad \tilde{\mu}_{\lambda_i} = \tilde{\Sigma}_{\lambda_i} (\sigma^{-2} Z_i'(Y_i - X_i \beta - \sum_{j=1}^J I_j W_j \gamma_j)) \]

where \( I_j \) is a \( J \times J \) diagonal matrix in which the \( j \)th entry is one, and zeros elsewhere.

3) \( p(\gamma_j | \text{rest}) = \mathcal{N}(\tilde{\mu}_{\gamma_j}, \tilde{\Sigma}_{\gamma_j}) \)

\[ \tilde{\Sigma}_{\gamma_j}^{-1} = \Gamma^{-1} + \sigma^{-2} W_j'W_j \quad \text{and} \quad \tilde{\mu}_{\gamma_j} = \tilde{\Sigma}_{\gamma_j} (\sigma^{-2} W_j'(Y_j - X_j \beta - \sum_{i=1}^I I_i Z_j \lambda_i)) \]

where \( I_i \) is a \( I \times I \) diagonal matrix in which the \( i \)th entry is one, and zeros elsewhere.

4) \( p(\Lambda | \text{rest}) = \mathcal{IW}(\tilde{\rho}_\Lambda, \tilde{R}_\Lambda) \)

\[ \tilde{\rho}_\Lambda = \rho_\Lambda + I \quad \text{and} \quad \tilde{R}_\Lambda = R_\Lambda + \sum_{i=1}^I \lambda_i \lambda_i' \]

\[ \text{We choose to use the following parameterization for the inverse Wishart and inverse Gamma distributions} \]

\[ \mathcal{IW}(\Lambda | \rho_\Lambda, R_\Lambda) = \frac{|R_\Lambda|^{\rho_\Lambda/2}}{2^{\rho_\Lambda/4} \Gamma_{h_\Lambda}((\rho_\Lambda)/2)} |\Lambda|^{-(\rho_\Lambda + h_\Lambda + 1)/2} e^{-\text{Tr}(R_\Lambda \Lambda^{-1})/2} \]
\[ \mathcal{IG}(\sigma^2 | a, b) = \frac{b^a}{\Gamma(a)} (\sigma^2)^{-a-1} e^{-b/\sigma^2}. \]
5) \( p(\Gamma|\text{rest}) = \text{IW}(\tilde{\rho}_\Gamma, \tilde{R}_\Gamma) \)

\[ \tilde{\rho}_\Gamma = \rho_\Gamma + J \quad \text{and} \quad \tilde{R}_\Gamma = R_\Gamma + \sum_{j=1}^{J} \gamma'_j \]

6) \( p(\sigma^2|\text{rest}) = \text{IG}(\tilde{a}, \tilde{b}) \)

\[ \tilde{a} = a + N/2 \quad \text{and} \quad \tilde{b} = b + \frac{1}{2} \sum_{i=1}^{I} \sum_{j=1}^{J} (y_{ij} - x'_{ij}\beta - z'_{ij}\lambda_i - w'_{ij}\gamma_j)^2 \]

Mean-field approximation assumes

\[ q(\beta, \{\lambda_i\}_i, \{\gamma_j\}_j, \Lambda, \Gamma, \sigma^2) = q(\beta)q(\Lambda)q(\Gamma)q(\sigma^2) \prod_{i=1}^{M} q(\lambda_i) \prod_{j=1}^{J} q(\gamma_j) \quad (32) \]

The variational distributions are

1) \( q(\beta) \propto \exp\{E_{\text{rest}}[\log p(\beta|\text{rest})]\} \Rightarrow q(\beta) = N(\mu_{q(\beta)}, \Sigma_{q(\beta)}) \), where

\[ \Sigma^{-1}_{q(\beta)} = \Sigma^{-1}_\beta + a_{q(\sigma^2)}/b_{q(\sigma^2)}X'X \]

\[ \mu_{q(\beta)} = \Sigma_{q(\beta)}(\Sigma^{-1}_\beta \mu_\beta + a_{q(\sigma^2)}/b_{q(\sigma^2)}(X'Y - \sum_{i=1}^{I} X'_iZ_i\mu_{q(\lambda_i)} - \sum_{j=1}^{J} X'_jW_j\mu_{q(\gamma_j)})) \]

2) \( q(\lambda_i) \propto \exp\{E_{\text{rest}}[\log p(\lambda_i|\text{rest})]\} \Rightarrow q(\lambda_i) = N(\mu_{q(\lambda_i)}, \Sigma_{q(\lambda_i)}) \), where

\[ \Sigma^{-1}_{q(\lambda_i)} = \rho_{q(\lambda_i)}R^{-1}_{q(\lambda_i)} + a_{q(\sigma^2)}/b_{q(\sigma^2)}Z'_iZ_i \]

\[ \mu_{q(\lambda_i)} = \Sigma_{q(\lambda_i)}(a_{q(\sigma^2)}/b_{q(\sigma^2)}Z'_i(Y_i - X_i\mu_{q(\beta)} - \sum_{j=1}^{J} I_jW_j\mu_{q(\gamma_j)})) \]

3) \( q(\gamma_j) \propto \exp\{E_{\text{rest}}[\log p(\gamma_j|\text{rest})]\} \Rightarrow q(\gamma_j) = N(\mu_{q(\gamma_j)}, \Sigma_{q(\gamma_j)}) \), where

\[ \Sigma^{-1}_{q(\gamma_j)} = \rho_{q(\gamma_j)}R^{-1}_{q(\gamma_j)} + a_{q(\sigma^2)}/b_{q(\sigma^2)}W'_jW_j \]

\[ \mu_{q(\gamma_j)} = \Sigma_{q(\gamma_j)}(a_{q(\sigma^2)}/b_{q(\sigma^2)}W'_j(Y_j - X_j\mu_{q(\beta)} - \sum_{i=1}^{I} I_iZ_i\mu_{q(\lambda_i)})) \]

4) \( q(\Lambda) \propto \exp\{E_{\text{rest}}[\log p(\Lambda|\text{rest})]\} \Rightarrow q(\Lambda) = \text{IW}(\rho_{q(\Lambda)}, R_{q(\Lambda)}) \), where

\[ \rho_{q(\Lambda)} = \rho_\Lambda + I \quad \text{and} \quad R_{q(\Lambda)} = R_\Lambda + \sum_{i=1}^{I} (\Sigma_{q(\lambda_i)} + \mu_{q(\lambda_i)}\mu_{q(\lambda_i)}') \]
5) $q(\Gamma) \propto \exp\{E_{\text{rest}}[\log p(\Gamma|\text{rest})]\} \Rightarrow q(\Gamma) = IW(\rho_q(\Gamma), R_q(\Gamma))$, where

$$\rho_q(\Gamma) = \rho_\Gamma + J \quad \text{and} \quad R_q(\Gamma) = R_\Gamma + \sum_{j=1}^J (\Sigma_{q(\gamma_j)} + \mu_{q(\gamma_j)} \mu_{q(\gamma_j)}')$$

6) $q(\sigma^2) \propto \exp\{E_{\text{rest}}[\log p(\sigma^2|\text{rest})]\} \Rightarrow q(\sigma^2) = IG(a_{q(\sigma^2)}, b_{q(\sigma^2)})$, where

$$a_{q(\sigma^2)} = a + N/2$$

$$b_{q(\sigma^2)} = b + \frac{1}{2} \sum_{i=1}^J \left( ||Y_i - X_i \mu_{q(\beta)} - Z_i \mu_{q(\lambda_i)} - \sum_{j=1}^J I_j W_i \mu_{q(\gamma_j)}||^2 \right.$$

$$+ \text{Tr}[X_i' X_i \Sigma_{q(\beta)}] + \text{Tr}[Z_i' Z_i \Sigma_{q(\lambda_i)}]) + \frac{1}{2} \text{Tr}[\sum_{j=1}^J W_j' W_j \Sigma_{q(\gamma_j)}].$$

The evidence lower bound is

$$\mathcal{L} = - (\mu_{q(\beta)} - \mu_{q(\beta)})' \Sigma_{q(\beta)}^{-1} (\mu_{q(\beta)} - \mu_{q(\beta)}) - \text{Tr}[\Sigma_{q(\beta)}^{-1} \Sigma_{q(\gamma)}]$$

$$- (\rho_{\Lambda} + J) \log |R_{q(\Lambda)}| - \rho_{q(\Lambda)} \sum_{i=1}^I \mu_{q(\Lambda_i)}' R_{q(\Lambda)}^{-1} \mu_{q(\Lambda_i)} - \rho_{q(\Lambda)} \text{Tr}[R_{q(\Lambda)}^{-1} (R_{\Lambda} + \sum_{i=1}^I \Sigma_{q(\Lambda_i)})]$$

$$- (\rho_{\Gamma} + J) \log |R_{q(\Gamma)}| - \rho_{q(\Gamma)} \sum_{j=1}^J \mu_{q(\gamma_j)}' R_{q(\Gamma)}^{-1} \mu_{q(\gamma_j)} - \rho_{q(\Gamma)} \text{Tr}[R_{q(\Gamma)}^{-1} (R_{\Gamma} + \sum_{j=1}^J \Sigma_{q(\gamma_j)})]$$

$$+ \log |\Sigma_{q(\beta)}| + \sum_{i=1}^I \log |\Sigma_{q(\Lambda_i)}| + \sum_{j=1}^J \log |\Sigma_{q(\gamma_j)}| - 2a_{q(\sigma^2)} \log b_{q(\sigma^2)}$$

**B Derivatives of the Ordinal Logit Model**

Conditional on the model parameters, ordered choices are independent across observations. For brevity here we show the gradient and Hessian based on a single choice observation, i.e., for a particular $i$ and a particular $j$. Later we can write down these derivatives for the whole model simply by summing up the results appropriately across observations.

Suppose the $L + 2$ ordered categories are $r_0, r_1, \ldots, r_{L+1}$. Denote

$$P_l = \frac{\exp(\tau_l - x_{ij} \lambda_i)}{1 + \exp(\tau_l - x_{ij} \lambda_i)} \quad \text{for} \quad l = 1, 2, \ldots, L,$$

and

$$P_0 = \frac{\exp(-x_{ij} \lambda_i)}{1 + \exp(-x_{ij} \lambda_i)} \quad \text{and} \quad P_{L+1} = 1 \quad \text{and} \quad P_{-1} = 0.$$
Then, on choice occasion \( j \), the probability for individual \( i \) choosing a category \( l \) can be written as 
\[ p(r_{ij}) = P_l - P_{l-1}, \text{ for } l = 0, 1, \ldots, L + 1. \]

The logarithm of the likelihood function of the single observation is 
\[
\log p(y_{ij} | \lambda_i, \kappa) = \sum_{h=0}^{L+1} d_{ijh} \log \left( P_l - P_{l-1} \right),
\]
where \( d_{ijh} \) is a binary indicator on whether individual \( i \) chose item \( h \) or not on occasion \( j \).

Note the useful facts that
\[
\frac{\partial P_l}{\partial \lambda_i} = P_l(P_l - 1)x_{ij}
\quad \text{and} \quad
\frac{\partial P_l}{\partial \tau_l} = P_l(1 - P_l).
\]

The gradient vector with respect to \( \lambda_i \) is
\[
\frac{\partial \log p(y_{ij}, \lambda_i, \kappa)}{\partial \lambda_i} = \sum_{l=0}^{L+1} d_{ijl}(P_{ijl} + P_{ij,l-1} - 1)x_{ij} - \rho_{q(\lambda)}R_{q(\lambda)}^{-1}(\lambda_i - \mu_{q(\beta)})
\]

The Hessian matrix with respect to \( \lambda_i \) is
\[
\frac{\partial^2 \log p(y_{ij}, \lambda_i, \kappa)}{\partial \lambda_i^2} = \sum_{l=0}^{L+1} d_{ijl}(P_l(P_l - 1) + P_{l-1}(P_{l-1} - 1))x_{ij}x_{ij} - \rho_{q(\lambda)}R_{q(\lambda)}^{-1}
\]

The derivatives for cutoff parameters are more complicated than those for the \( \lambda_i \)'s. Denote two scalar quantities,
\[
s_1 = \frac{P_l(1 - P_l)}{P_l - P_{l-1}} \quad \text{and} \quad s_2 = \frac{P_{l-1}(1 - P_{l-1})}{P_l - P_{l-1}},
\]

After the reparameterization from \( \tau \) to \( \kappa \), we can write down the gradient vector with respect to \( \kappa \) as
\[
\frac{\partial \log p(y_{ij}, \lambda_i, \kappa)}{\partial \kappa} = \sum_{l=1}^{L+1} d_{ijkl}(s_1 \frac{\partial \tau_l}{\partial \kappa} - s_2 \frac{\partial \tau_{l-1}}{\partial \kappa}) - \Sigma_{-1}^{-1}(\kappa - \mu_{\kappa}),
\]

where \( \frac{\partial \tau_l}{\partial \kappa} = (e^{\kappa_1}, \ldots, e^{\kappa_l}, 0, \ldots, 0)' \), for \( l = 1, \ldots, L \).

The Hessian matrix with respect to \( \kappa \) is
\[
\frac{\partial^2 \log p(y_{ij}, \lambda_i, \kappa)}{\partial \kappa^2} = \sum_{l=1}^{L+1} d_{ijkl} \left( s_1 \frac{\partial^2 \tau_l}{\partial \kappa \partial \kappa'} - s_2 \frac{\partial^2 \tau_{l-1}}{\partial \kappa \partial \kappa'} + s_1(1 - 2P_l - s_1)\frac{\partial \tau_l \partial \tau_l}{\partial \kappa \partial \kappa'} \right.
\]
\[
- s_2(1 - 2P_{l-1} + s_2)\frac{\partial \tau_{l-1} \partial \tau_{l-1}}{\partial \kappa \partial \kappa'} + s_1s_2\frac{\partial \tau_{l-1} \partial \tau_l}{\partial \kappa \partial \kappa'} + s_1s_2\frac{\partial \tau_l \partial \tau_{l-1}}{\partial \kappa \partial \kappa'} \left) - \Sigma_{-1}^{-1},
\]

where \( \frac{\partial^2 \tau_l}{\partial \kappa \partial \kappa'} \) is a diagonal matrix with the vector \( \frac{\partial \tau_l}{\partial \kappa} \) on the diagonal.