Fast variational inference for multinomial probit models

Rubén Loaiza-Maya and Didier Nibbering∗

Department of Econometrics and Business Statistics, Monash University

February 28, 2022

Abstract

The multinomial probit model is often used to analyze choice behaviour. However, estimation with existing Markov Chain Monte Carlo (MCMC) methods is computationally costly, which limits its applicability to large choice data sets. This paper proposes a variational inference method that is fast, even when a large number of choice alternatives and observations are considered. Variational methods usually require an analytical expression for the unnormalized posterior density and an adequate choice of variational family. Both are challenging to specify in a multinomial probit, which has a posterior that requires identifying restrictions and is augmented with a large set of latent utilities. We employ a spherical transformation on the covariance matrix of the latent utilities to construct an unnormalized augmented posterior that identifies the parameters, and use the conditional posterior of the latent utilities as part of the variational family. The proposed method is faster than MCMC, and can be made scalable to both a large numbers of choice alternatives and a large number of observations. The accuracy and scalability of our method is illustrated in numerical experiments and real purchase data with one million observations.

Keywords: Multinomial probit model, Variational inference, Large choice data sets

JEL Classification: C11, C35, C55, C61

∗Correspondence to: Department of Econometrics & Business Statistics, Monash University, Clayton VIC 3800, Australia, e-mail: didier.nibbering@monash.edu
1 Introduction

The multinomial probit (MNP) model is a popular tool for analyzing choice behavior, with recent applications including brand choices (Miyazaki et al., 2021), employment choices (Mishkin, 2021), and car parking choices (Paleti, 2018). The main advantage of the MNP model is the relaxation of the independence of irrelevant alternatives property of multinomial logit models. The MNP achieves this by specifying the conditional covariance matrix of the latent utilities of the choice alternatives. However, estimation of the MNP model is computationally costly. The evaluation of the likelihood function involves high-dimensional integrals, which can be solved via simulation methods such as Markov chain Monte Carlo (MCMC). In this paper we propose a variational inference (VI) method for estimation in the MNP model, which is fast even when applied to choice sets that have a large number of choice alternatives and a large number of observations.

Developments in the Bayesian analysis of the MNP model has reduced the computational complexity of parameter estimation, but its applicability to modern choice data sets is still limited. Bayesian estimation of the MNP model augments the likelihood function with a set of latent utilities, which are then generated inside an MCMC scheme (Albert and Chib, 1993). This approach avoids the computationally costly step of directly calculating the choice probabilities in the likelihood function via numerical integration. Instead, each MCMC iteration draws a vector of latent utilities from a truncated normal distribution (McCulloch and Rossi, 1994) for each observation. Since these draws are highly auto-correlated, a large amount of iterations are required to achieve convergence. Therefore, this approach still has a substantial computational burden, especially when the number of observations or the number of choice alternatives is large.

VI is a computationally scalable alternative to MCMC. Instead of sampling from the posterior, VI calibrates a parametric approximating density by minimizing the distance to the posterior. Applying VI to the MNP model poses two main challenges. First, parameter identification in the MNP model requires a restriction on the covariance matrix (Bunch, 1991). Burgette and Nordheim (2012) show that a trace restriction has the best performance, and introduce a “working parameter” that rescales the trace within the MCMC algorithm. This parameter is not part of the model specification and not identified given the data. Hence an analytical expression for the unnormalized posterior density, which is required for the implementation of VI, is not available. Second, to reduce its computational complexity, the posterior density of the MNP model has to be augmented with a large number of latent variables. Existing VI methods for latent variable models often make strong assumptions on the approximating densities for the latent variables (Westling and McCormick, 2019).

Because of these challenges, existing VI approaches are designed for restrictive specifications of the MNP model. For instance, Girolami and Rogers (2006) model the latent utilities as independent Gaussian processes, that only allow for correlations that are a function of observed regressors. Fasano and Durante (2022) conduct variational inference on the posterior for the coefficients conditional on a fixed covariance matrix. Moreover, both papers impose strong inde-
dependence assumptions on the family of variational approximations to the posterior density.

This paper proposes a VI method for the MNP model that overcomes the two challenges. First, we construct an analytical expression of the unnormalized augmented posterior density by using the MNP model specification as proposed by Loaiza-Maya and Nibbering (2021). They transform the covariance parameters into a spherical coordinate system. The spherical transformation naturally imposes the trace restriction on the covariance matrix, and therefore the parameters are identified within the model specification. Second, we construct an accurate approximation to the augmented posterior by applying the methods in Loaiza-Maya et al. (2021), who develop accurate variational approximations for models with multiple latent variables. The approximation for the latent utilities is the exact conditional posterior distribution for the latent utilities, and the approximation for the coefficients and the parameters in the spherical transformation is Gaussian. The combination of these two innovations results in a VI approach that is substantially faster than MCMC.

Additionally, we demonstrate that our VI approach is scalable to data sets with a large number of observations. The VI optimization problem is solved with stochastic gradient ascent (SGA), where each iteration takes a draw from the conditional posterior distribution for the latent utilities. Sampling from the conditional posterior distribution of the latent utilities is computationally costly and hence takes the majority of the computation time in VI. In contrast to MCMC, SGA allows for subsampling, which means that in each iteration only a subsample of the latent utilities have to be generated. This further reduces the computational complexity of the proposed methods.

We formulate our VI approach for the general multivariate multinomial probit (MVMNP) model. This means that we provide one common method for fast inference for a variety of different models, such as the MNP and multivariate probit (MVP) models, that are currently estimated with different identification strategies and different MCMC methods. For instance, Zhang et al. (2006) and Talhouk et al. (2012) fix the covariance matrix to be a correlation matrix in an MVP model. Chib et al. (1998) and McCulloch et al. (2000) fix one element of the covariance matrix in MNP models, and Zhang et al. (2008) extends this to the MVMNP model. Burgette and Nordheim (2012) introduce the trace restriction in the MNP model, and Richard Hahn et al. (2012) fix the scale of a factor structure in the covariance matrix in the MVP model. Moreover, our method extends the spherical transformation on the covariance parameters to any MVMNP model, which allows for a factor structure in the covariance matrix that can substantially reduce the number of parameters to be estimated. Hence, our method can be applied to data sets with a large number of choices and a large number of choice alternatives.

Numerical experiments show that our VI method provides accurate parameter estimates and predictions, while it only takes a fraction of the computational cost of MCMC. A numerical experiment with a small data set of 10,000 observations, in which MCMC is feasible, shows a minimal loss in accuracy of VI relative to MCMC. An experiment with one million observations, in which MCMC is infeasible, shows that VI applied to a large data set can improve in accuracy relative to MCMC applied to only a subset of the observations.
We illustrate the practical relevance of our method with two empirical applications. First, VI produces similar results as MCMC within 10% of the computation time, in a small real data set of laundry detergent purchases. The second application considers a large-scale choice data set. Data on large choice sets with a large number of observations is nowadays available in many applications of MNP models. MCMC in the MNP model is considered to be infeasible when the number of observations is large. We estimate the MNP model with the proposed VI method on more than a million pasta purchases in less than 1.5 hours.

The outline of the remainder of this paper is as follows. Section 2 discusses the MVMNP model specification and Section 3 develops our VI method. Section 4 conducts numerical experiments to evaluate estimation and predictive accuracy and computational costs, and Section 5 applies the proposed methods to two real consumer choice data sets. Section 6 concludes.

2 Model specification

2.1 Multivariate multinomial probit model

We observe $K$ multinomial choices, where each choice $k = 1, \ldots, K$ has $J_k + 1$ choice alternatives, for individual $i = 1, \ldots, N$. Let $\mathbf{Y}_i = (Y_{i1}, \ldots, Y_{iK})^\top$ denote the $K$-dimensional random variable describing the joint set of choices for individual $i$, where $Y_{ik} = j$ if individual $i$ chooses $j = 0, 1, \ldots, J_k$ for the $k$-th choice. The number of potential outcomes of $\mathbf{Y}_i$ is $\prod_{k=1}^K (J_k + 1)$.

Assume that for the $k$-th choice there is a $J_k$-dimensional vector $\mathbf{Z}_{ik} = (Z_{ik1}, \ldots, Z_{ikJ_k})^\top$ of continuous random variables representing the latent utilities for the choice alternatives, and which excludes the base-category latent utility $Z_{ik0}$. The multinomial outcome $Y_{ik}$ is determined by the maximum value of $\mathbf{Z}_{ik}$ as follows:

$$Y_{ik} = \begin{cases} 0 & \text{if } \max(\mathbf{Z}_{ik}) < 0, \\ j & \text{if } Z_{ikj} = \max(\mathbf{Z}_{ik}) > 0, \end{cases}$$

where $\max(\mathbf{Z}_{ik})$ is the largest element of $\mathbf{Z}_{ik}$. The latent utilities corresponding to the choice alternatives in choice $k$ are modeled as

$$\mathbf{Z}_{ik} = X_{ik} \mathbf{\beta}_k + \mathbf{\epsilon}_{ik},$$

where $X_{ik}$ is a $J_k \times r_k$ regressor matrix, $\mathbf{\beta}_k$ is an $r_k$-dimensional vector of coefficients and $\mathbf{\epsilon}_{ik} = (\epsilon_{ik1}, \ldots, \epsilon_{ikJ_k})^\top$ is a $J_k$-dimensional disturbance vector with mean zero.

The regressor matrix $X_{ik}$ typically includes $J_k$ choice alternative-specific intercepts, an $n_d$-dimensional vector $\mathbf{x}_{i,d}$ of individual-specific characteristics, and a $(J_k + 1) \times n_a$ matrix $X_{ik,a}$ of
alternative-specific covariates, such that \( r_k = J_k + J_k n_d + n_a \) and

\[
X_{ik} = [I_{J_k} \ \mathbf{x}_{id}^\top \otimes I_{J_k} \ T_k X_{ik,a}],
\]

(3)

with transformation matrix \( T_k = [-J_k I_{J_k}] \), where \( J_k \) denotes the \( J_k \)-dimensional vector of ones and \( I_{J_k} \) the \( J_k \times J_k \) identity matrix.

The model specified in (1) and (2) only considers the \( J_k \) utilities indicated by \( Z_{ik} \). Therefore, the covariates in the regressor matrix \( X_{ik} \) in (3) are transformed to match the dimensions of \( Z_{ik} \).

The transformation matrix \( T_k \) subtracts the covariates corresponding to choice category \( j = 0 \) in choice \( k \) from the covariates corresponding to the remaining choice alternatives in choice \( k \). This model specification addresses the first parameter identification problem in the MVMNP model: additive redundancy arises if a unique utility is specified for each choice alternative, as discussed in Bunch (1991). The second identification problem is caused by multiplicative redundancy: multiplying both sides of (2) by a positive scalar does not change \( Y_{ik} \) in (1). We address this identification problem in Section 2.3.

By specifying a joint distribution for \( \varepsilon_i = (\varepsilon_{i1}^\top, \ldots, \varepsilon_{iK}^\top)^\top \), the MVMNP model can allow for within-choice and between-choice correlation in the latent utilities. The \( K \) models implied by (2) can be stacked to obtain the multivariate utility model

\[
Z_i = X_i \beta + \varepsilon_i, \quad \varepsilon_i \sim N[0_J, \Sigma],
\]

(4)

where \( Z_i = (Z_{i1}^\top, \ldots, Z_{iK}^\top)^\top \) is a \( J \)-dimensional vector with \( J = \sum_{k=1}^K J_k \), \( X_i = \text{blockdiag}(X_{i1}, \ldots, X_{iK}) \) is a \( J \times r \) block diagonal regressor matrix with \( r = \sum_{k=1}^K r_k \), and \( \beta = (\beta_1^\top, \ldots, \beta_K^\top)^\top \) is an \( r \)-dimensional vector of coefficients. The \( J \times J \) covariance matrix of the disturbance vector \( \varepsilon_i \) can be represented as

\[
\Sigma = \begin{bmatrix}
\Sigma_{11} & \Sigma_{12} & \cdots & \Sigma_{1K} \\
\Sigma_{21} & \Sigma_{22} & \cdots & \Sigma_{2K} \\
\vdots & \vdots & \ddots & \vdots \\
\Sigma_{K1} & \Sigma_{K2} & \cdots & \Sigma_{KK}
\end{bmatrix},
\]

(5)

where \( \Sigma_{kl} = \Sigma_{lk} \). The \( J_k \times J_l \) covariance matrix \( \Sigma_{kl} = \text{cov}(\varepsilon_{ik}, \varepsilon_{il}) \) captures the correlation across the utilities within-choice \( k \) if \( k = l \), and the correlation across the utilities between choices \( k \) and \( l \) if \( k \neq l \), with \( k = 1, \ldots, K \) and \( l = 1, \ldots, K \).

The MVMNP model easily simplifies to other commonly used choice models. First, for \( K = 1 \) the MVMNP boils down to a multinomial probit (MNP) model, with \( J + 1 \) choice alternatives corresponding to potentially correlated latent utilities. When \( K > 1 \) and the elements of \( \Sigma_{lk} \) equal zero for all \( k \neq l \) in (3), we have \( K \) independent MNP models. Second, for \( J = 1 \) we have a multivariate probit (MVP) model, with \( K \) potentially correlated binary choices. When \( K = 1 \) and \( J = 1 \), the model boils down to a simple binary probit model.
For the remainder of the paper we stack the regressor matrices for all individuals in the \((NJ) \times r\) matrix \(X = [X_1^\top | \ldots | X_N^\top]^\top\), the random choice vectors in \(Y = (Y_1^\top, \ldots, Y_N^\top)^\top\), and the random latent utility vectors in \(Z = (Z_1^\top, \ldots, Z_N^\top)^\top\).

2.2 Factor structure covariance matrix

The total number of unique parameters in the covariance matrix \(\Sigma\) equals \(J(J+1)/2\), which grows quadratically in the number of choices and the number of choice alternatives. Since these parameters have to be estimated from a single multinomial variable \(Y_i\), accurate parameter estimation is challenging if either \(J\) or \(K\) is large, or both, even when a relatively large number of observations \(N\) is available.

To reduce the dimension of the parameter space, we specify a factor structure for \(\Sigma\). Define the \(J \times p\) matrix \(B\) with \(p < J\) and the \(J \times J\) diagonal matrix \(D\). We model \(\Sigma\) as

\[
\Sigma = BB^\top + D^2. \tag{6}
\]

The total number of parameters that characterize the covariance matrix in (6) is \(n = J(p + 1)\). This implies that for a given value of \(p\), the number of parameters grows linearly with \(J\), instead of quadratically.

To understand the implications of this factor structure on the within- and between-choice correlations, the matrices \(B\) and \(D\) are partitioned as \(B = [B_1^\top | \ldots | B_K^\top]^\top\) and \(D = \text{diag}(d_1^\top, \ldots, d_K^\top)\), where \(B_k\) is a \(J_k \times p\) matrix and \(d_k\) a \(J_k\)-dimensional vector corresponding to choice \(k\). The within-choice covariance matrix \(\Sigma_{kk}\) can then be expressed as

\[
\Sigma_{kk} = B_kB_k^\top + D_k^2, \tag{7}
\]

which shows that the matrices \(\{\Sigma_{kk}\}_{k=1}^K\), and hence the within-choice correlations for each choice, are characterised by disjoint sets of model parameters. The between-choice covariance matrix \(\Sigma_{kl}\) with \(k \neq l\) is given by

\[
\Sigma_{kl} = B_kB_l^\top, \tag{8}
\]

which is only a function of the matrices \(B_k\) and \(B_l\) corresponding to the within-choice correlations in choices \(k\) and \(l\). In sum, the matrices \(\{B_k\}_{k=1}^K\) determine both the within- and between-choice covariances in the latent utilities.

2.3 Parameter identification

As discussed in Section 2.1, the scale of the latent utilities is unidentified. To identify the model parameters, we extend the approach of Loaiza-Maya and Nibbering (2021) from an MNP model where \(K = 1\), to the MVMNP model with \(K > 1\). For each choice \(k\), we fix the scale with a trace
restriction \(\text{trace}(\Sigma_{kk}) = J_k\), by transforming the elements of \(B_k\) and \(d_k\) into a spherical coordinate system.

This parameter identification strategy has three main advantages. First, in contrast to alternative identification restrictions, the trace restriction identifies the model parameters without fixing specific elements in the covariance matrix \(\Sigma\). Burgette and Nordheim (2012) show that Bayesian estimation in the MNP model is sensitive to which elements in the covariance matrix are fixed. Second, the spherical transformation on the factor covariance structure simplifies parameter estimation, as it naturally satisfies the trace restriction. Instead of performing inference on a parameter space with a joint restriction on all elements in each \(\Sigma_{kk}\), we perform inference on a parameter space for which no joint parameter restriction is required. Third, since the spherical transformation naturally imposes the trace restriction, our approach does not require rescaling of the covariance matrix. Therefore, there is an analytical expression available for the augmented posterior \(p(\theta, z | y, X)\) which allows us to apply variational inference. Even with \(p = J\), in which case there is no dimension reduction, writing \(\Sigma\) as in (6) has the benefit that it can be transformed by a spherical transformation and VI can be applied.

The spherical transformation is applied to the vector \(\psi_k\), which is constructed from the elements of \(B_k\) and \(d_k\) as

\[
\psi_k = (\psi_{k1}, \ldots, \psi_{kn_k})^\top = (\text{vec}(B_k)^\top, d_k^\top)^\top,
\]

where \(n_k = J_k(p + 1)\), \(\text{vec}(\cdot)\) denotes the vectorization operator, and \(\text{trace}(\Sigma_{kk}) = \sum_{i=1}^{n_k} \psi_{ki}^2\). We transform \(\psi_k\) into a spherical coordinate system that is defined by a radius, which we set to \(\sqrt{J_k}\), and an \((n_k - 1)\)-dimensional vector of angles \(\kappa_k = (\kappa_{k1}, \ldots, \kappa_{kn_k-1})^\top\).

The spherical transformation reparameterises \(\psi_k\) in terms of \(\kappa_k\) as

\[
\psi_{kl}(\kappa_k) = \begin{cases} 
\sqrt{J_k \cos \kappa_{kl}} & \text{for } l = 1, \\
\sqrt{J_k \cos \kappa_{kl} \prod_{j=1}^{l-1} \sin \kappa_{kj}} & \text{for } 1 < l < n_k, \\
\sqrt{J_k \prod_{j=1}^{n_k-1} \sin \kappa_{kj}} & \text{for } l = n_k,
\end{cases}
\]

where \(\kappa_{kl} \in [0, \pi)\) for \(l < n_k - J_k + 1\). The remaining angle bounds \(\kappa_{kl} \in [0, \pi/2)\) for \(n_k - J_k + 1 \leq l \leq n_k - 1\) ensure that the elements of \(d_k\) are strictly greater than zero. This ensures that the map from \(d_k\) to \(\Sigma_{kk}\) in (7) is bijective.

By construction, the transformation in (10) satisfies \(\sum_{l=1}^{n_k} \psi_{kl}(\kappa_k)^2 = J_k\) for any value of \(\kappa_k\). This reparametrization is applied to all \(\Sigma_{kk}\), with \(k = 1, \ldots, K\), which results in a covariance matrix \(\Sigma\) that is fully characterised by the \(n\)-dimensional parameter vector \(\kappa = (\kappa_1^\top, \ldots, \kappa_K^\top)^\top\), with \(n = J(p + 1)\). The parameter vector \(\kappa\) imposes \(K\) trace restrictions simultaneously: one for each choice.
The inverse function of the spherical transformation in (10) is
\[
\kappa_{kl}(\psi_k) = \begin{cases} 
\arccos \left( \psi_{kl} \left( \sum_{j=l}^{n_k} \psi_{kj}^2 \right)^{-\frac{1}{2}} \right) & \text{for } l < n_k - 1, \\
\arccos \left( \psi_{kl} \left( \sum_{j=l}^{n_k} \psi_{kj}^2 \right)^{-\frac{1}{2}} \right) & \text{for } \{ l = n_k - 1 \land \psi_{kn_k} \geq 0 \}, \\
2\pi - \arccos \left( \psi_{kl} \left( \sum_{j=l}^{n_k} \psi_{kj}^2 \right)^{-\frac{1}{2}} \right) & \text{for } \{ l = n_k - 1 \land \psi_{kn_k} < 0 \},
\end{cases}
\] (11)
where \( \kappa_{kl} = 0 \) if \( \psi_{kl} > 0 \) and \( \psi_{k,l+1} = \cdots = \psi_{kn_k} = 0 \), and \( \kappa_{kl} = \pi \) if \( \psi_{kl} < 0 \) and \( \psi_{k,l+1} = \cdots = \psi_{kn_k} = 0 \).

3 Bayesian estimation

This section develops a Bayesian method for estimating the parameters in the \( m \)-dimensional vector \( \theta = (\beta^T, \kappa^T)^T \), with \( m = r + n \). We conduct inference of the augmented posterior distribution
\[
p(\theta, z|y, X) \propto p(y, z|X, \theta)p(\theta),
\] (12)
where we use lower case letters to denote realised values of the corresponding random vectors. For instance, \( y \) is the realised vector of \( Y \).

The augmented likelihood function is given by
\[
p(y, z|X, \theta) = p(y|z)p(z|X, \theta) = \prod_{i=1}^{N} p(y_i|z_i) \phi_J(z_i; X_i\beta, \Sigma(\kappa)),
\] (13)
where \( \phi_J(z_i; X_i\beta, \Sigma(\kappa)) \) denotes the \( J \)-variate normal density with mean \( X_i\beta \) and covariance matrix \( \Sigma(\kappa) \), with \( \Sigma(\kappa) \) the covariance matrix constructed from the vector of angles \( \kappa \), \( p(y_i|z_i) = \prod_{k=1}^{K} p(y_{ik}|z_{ik}) \) and
\[
p(y_{ik}|z_{ik}) = \begin{cases} 
I[z_{ik}y_i = \max(z_{ik})] & \text{if } \max(z_{ik}) > 0, \\
I(y_{ik} = 0) & \text{if } \max(z_{ik}) \leq 0,
\end{cases}
\] (14)
where \( I[A] \) is an indicator function that equals one if \( A \) is true and zero otherwise.

We set the prior distribution as \( p(\theta) = p(\beta) \prod_{k=1}^{K} p(\kappa_k) \), with \( \beta \sim N(0, \frac{1}{10}I_r) \) and \( p(\kappa_k) \) specified in Appendix A.

3.1 Markov Chain Monte Carlo sampling

The posterior density in (12) can be computed using Markov Chain Monte Carlo sampling. For each individual, the latent utility of each choice alternative is sampled conditional on all the other choice alternatives from a truncated normal. This process induces a sequence of latent utility
draws that is highly auto-correlated. Therefore MCMC requires a large number of iterations such that convergence is achieved. Since each iteration involves \( N \times J \) draws from a truncated normal, MCMC is computationally costly. Appendix B derives the MCMC sampling scheme for the MVMNP model.

The computational costs of an MCMC sampling scheme increase in the number of choice alternatives \( J_k \) in each choice \( k \), the number of choices \( K \), and the number of observations \( N \). As a result, when the total number of choice alternatives \( J \) is large, MCMC is considered to be computationally practical as long as the number of observations is small. There are two empirical settings where this is the case. First, univariate choice sets \((K = 1)\) that have a large number of choice alternatives. Second, applications that consider multiple choices and for which the overall number of choice alternatives \( J \) is large. However, it is precisely in these type of settings that having a large number of observations is key for accurate estimation of the high-dimensional covariance matrix of the latent utilities.

### 3.2 Variational inference

To circumvent the computational challenges of MCMC, we utilize variational inference. VI approximates the posterior density in (12) by a parametric density \( q_\lambda(\theta, z) \in Q \) from the class of density functions \( Q = \{ q_\lambda(\theta, z) : \lambda \in \Lambda \} \), where \( q_\lambda(\theta, z) \) is indexed by the variational parameter vector \( \lambda \in \Lambda \). The optimal variational parameter \( \hat{\lambda} \) is obtained by maximizing the evidence lower bound (ELBO) function \( L(\lambda) = \mathbb{E}_{q_\lambda}[\log g(\theta, z) - \log q_\lambda(\theta, z)] \):

\[
\hat{\lambda} = \arg \max_{\lambda \in \Lambda} \mathbb{E}_{q_\lambda}[\log g(\theta, z) - \log q_\lambda(\theta, z)],
\]

where \( g(\theta, z) = p(y|z)p(z|X, \theta)p(\theta) \) is the unnormalized posterior density. Ormerod and Wand (2010) show that the optimization problem in (15) is equivalent, but computationally more efficient, to minimizing the Kullback-Leibler (KL) divergence between \( q_\lambda(\theta, z) \) and the exact posterior density \( p(\theta, z|y, X) \).

Unlike MCMC, which generates draws from the exact posterior distribution, VI constructs a parametric approximation to it. As a result, VI has three main advantages over MCMC. First, VI relies on optimization instead of sampling, and therefore does not suffer from high auto-correlation between draws from the exact posterior. Second, VI requires much less storage memory as the output from VI is the calibrated parameter vector of the approximation, rather than a large number of parameter draws. Third, VI can readily incorporate subsampling of the latent utilities in the optimization routine, which can substantially reduce the computational burden.

#### 3.2.1 Variational family

Key to the implementation of VI is the choice of the variational family \( Q \). We set \( q_\lambda(\theta, z) = p(z|\theta, y, X)q_\lambda(\theta) \), where \( p(z|\theta, y, X) \) is the conditional posterior of the latent utilities defined
in Appendix C, and we define $q_\lambda(\theta)$ below. Loaiza-Maya et al. (2021) show that the optimization problem in (15) with this variational family is equivalent to the optimization problem that considers the KL divergence between the intractable posterior $p(\theta | y, X)$ and $q_\lambda(\theta)$. Alternative specifications for the variational family may result in approximating errors for the latent utilities, which can provide inconsistent estimates, as shown in Westling and McCormick (2019).

For the choice of $q_\lambda(\theta)$, we follow Ong et al. (2018) and employ a Gaussian density with mean $\mu$ and covariance matrix $\Omega = CC^\top + E^2$, where $C$ is a matrix of dimension $m \times s$ for $s < m$, $E = diag(e)$ and $e$ an $m$-dimensional vector. The variational parameter vector for this approximating class is $\lambda = (\mu^\top, \text{vech}(C)^\top, e^\top)^\top$, where the operator $\text{vech}$ denotes the half vectorization of a rectangular matrix.

### 3.2.2 Stochastic gradient ascent

We solve the optimization problem in (15) using stochastic gradient ascent (SGA) methods. SGA calibrates the variational parameter by iterating over

$$\lambda^{[j+1]} = \lambda^{[j]} + \rho^{[j]} \circ \nabla_{\lambda} \mathcal{L}(\lambda^{[j]}),$$

(16)

until convergence is achieved. The vector $\rho^{[j]}$ contains the so called “learning parameters”, which we set according to the ADADELTA approach in Zeiler (2012). The vector $\nabla_{\lambda} \mathcal{L}(\lambda^{[j]})$ is an unbiased estimate of the gradient of the ELBO evaluated at the iterate $\lambda^{[j]}$.

We construct $\nabla_{\lambda} \mathcal{L}(\lambda)$ using the following expression of the gradient

$$\nabla_{\lambda} \mathcal{L}(\lambda) = \mathbb{E}_{z, \xi} \left[ \frac{\partial \theta(\xi, \lambda)}{\partial \lambda}^\top \{ \nabla_\theta \log g(\theta(\xi, \lambda), z) - \nabla_\theta \log q_\lambda(\theta(\xi, \lambda)) \} \right],$$

(17)

where $\theta(\xi, \lambda) = \mu + C w + e \circ e$, $w \sim N(0_s, I_s)$, $e \sim N(0_m, I_m)$, and $\xi = (w^\top, e^\top)^\top$. This expression is derived in Loaiza-Maya et al. (2021) using the “re-parametrization trick” proposed in Kingma and Welling (2013). We derive $\nabla_\theta \log g(\theta(\xi, \lambda), z)$ for the MVMNP model in Appendix C and $\frac{\partial \theta(\xi, \lambda)}{\partial \lambda}$ and $\nabla_\theta \log q_\lambda(\theta(\xi, \lambda))$ are provided in Ong et al. (2018) for the variational family in Section 3.2.2.

An unbiased estimate of $\nabla_{\lambda} \mathcal{L}(\lambda)$ is constructed by using a sample estimate of the expectation. At each SGA iteration $[j]$, we calculate a sample estimate of (17) based on only one draw for both $\xi$ and $z$: $\xi^{[j]} \sim N(0_{s+m}, I_{s+m})$ and $z^{[j]} \sim p(z | \theta(\xi^{[j]}, \lambda^{[j]}), y, X)$. Since sampling directly from $p(z | \theta(\xi, \lambda), y, X)$ is infeasible, we generate the latent utility vector draw $z^{[j]}$ via $G$ Gibbs steps of the truncated normal algorithm proposed in McCulloch and Rossi (1994). The Gibbs sampling algorithm is started at the last iterate value $z^{[j-1]}$. Since these Gibbs draws are highly correlated, a larger value for $G$ increases the accuracy of the estimate for the gradient. However, each Gibbs step includes $N \times J$ draws from a truncated normal distribution, which is computationally costly. We find that $G = 10$ balances well accuracy and computationally speed.
3.2.3 Subsampling of the latent utilities

The objective of VI is to compute the optimal variational parameter vector, so it suffices to run enough SGA iterations until convergence is reached for all the elements of $\lambda^{[j]}$. SGA generally requires a small number of iterations, making it much faster than MCMC. However, the majority of the computation time is still spent on the generation of the latent utilities. Since SGA allows for subsampling of the observations, the computational burden of the latent utilities can be substantially reduced in VI.

Instead of sampling the latent utilities $z_i$ at each iteration for all individuals, SGA can estimate the gradient unbiasedly using only a subsample of the latent utilities. The ELBO gradient can be rewritten in terms of the variable $A \subset \{1, \ldots, N\} \sim f(A)$, where a draw from $f(A)$ is a random subsample of indexes without replacement. Define the subsample of latent utilities as $z_A = \{z_i\}_{i \in A}$. Since $E_A[\nabla_\theta \log g(\theta, z_A)] = \nabla_\theta \log g(\theta, z)$, it holds that

$$\nabla_\lambda L(\lambda) = E_{z, \zeta, A} \left[ \frac{\partial(\zeta, \lambda)^\top}{\partial \lambda} \{\nabla_\theta \log g(\theta(\zeta, \lambda), z_A) - \nabla_\theta \log q_\lambda(\theta(\zeta, \lambda))\} \right].$$

Appendix C provides the expression for $\nabla_\theta \log g(\theta, z_A)$ required to compute the subsampling gradient estimate for the MNP model. An unbiased estimate of (18) is constructed using a sample estimate of the expectation, using the draws $A^{[j]} \sim f(A)$, $\zeta^{[j]} \sim N(0_{s+m}, I_{s+m})$, and $z_A^{[j]} \sim p(z_A^{[j]}|\theta^{[j]}, \lambda^{[j]}, y, X)$.

3.2.4 Variational predictive distribution

The VI predictive probability mass function for $Y_i$ is given by

$$p_\lambda(Y_i|X_i) = \int p(Y_i|z_i)p(z_i|\theta, X_i)q_\lambda(\theta)\,d\theta\,dz_i,$$

where $X_i$ denotes the attributes of the observation $i$ to be predicted. We construct an estimate $\hat{p}_\lambda(Y_i|X_i)$ for (19) as the empirical probability mass implied by the draws $\{y_i^{[m]}\}_{m=1}^M$, which are obtained by drawing sequentially from $\theta^{[m]} \sim q_\lambda(\theta)$, $z_i^{[m]} \sim p(z_i|\theta^{[m]}, X_i)$ and $y_i^{[m]} \sim p(Y_i|z_i^{[m]})$.

To evaluate predictive performance, we employ the logarithmic score (log-score), which is a probabilistic measure of predictive accuracy. The choice-specific log-score is given as

$$\text{log-score}_k = \frac{1}{N} \sum_{i=1}^N \ln(\hat{p}_\lambda(Y_{ik}|X_{ik})), $$

and the total model fit can be assessed by the average log-scores across choices.

The point forecast $\hat{Y}_i$ for $Y_i$ is constructed as the mode of $\hat{p}_\lambda(Y_i|X_i)$. The point prediction
accuracy can be measured in terms of the hit-rate given as

\[ \text{hit-rate}_k = \frac{1}{N} \sum_{i=1}^{N} I[\hat{Y}_{ik} = Y_{ik}], \]

(21)

where \( I[A] \) is an indicator function. For both the hit-rate and the log-score large values are preferred.

4 Numerical experiments

This section presents two numerical experiments to assess the accuracy and the computational costs of the proposed VI approach. The first experiment compares VI to MCMC in a moderately sized data set in which MCMC is computationally feasible. The second experiment is on a large dataset for which VI estimation remains feasible, but MCMC does not.

4.1 Design

We generate a data set from the data generating process specified in (1) and (2) with \( K = 2 \) and \( J_1 = J_2 = 10 \). The elements of the matrices \( X_{i1,a} \) and \( X_{i2,a} \) are independently generated from normal distributions with corresponding mean \( \mu = 0 \) and variance \( \sigma^2 = 1 \). We do not include individual-specific characteristics \( x_{i,d} \).

The true parameter vector \( \beta_0 \) consists of \( J \) intercepts drawn independently from uniform distributions \( U(-0.5, 0) \), and the coefficients for \( X_{11,a} \) and \( X_{12,a} \) are respectively fixed at -0.5 and -0.8. The true covariance matrix \( \Sigma_0 \) is constructed from a factor structure with \( p = 10 \) for each choice. The elements of the factor structures are generated from normal distributions with corresponding mean \( \mu = 0 \) and variance \( \sigma^2 = 1 \). We consecutively apply the transformations in (9), (11), (10) and (6), to obtain the true covariance matrix \( \Sigma_0 \) with trace-restricted \( \Sigma_{11} \) and \( \Sigma_{22} \).

We apply our VI method to two generated datasets, one with \( N = 10,000 \) and the second one with \( N = 1,000,000 \). For both settings we generate an additional 10,000 observations for out-of-sample evaluation. VI with subsampling is denoted by VI(\( \frac{M}{N} \cdot 100\% \)), where \( \frac{M}{N} \cdot 100\% \) denotes the percentage of the total estimation sample \( N \) used in each VI iteration step. For the purpose of comparison, we also estimate a MVMNP model with the covariance matrix fixed at the identity matrix (VI(I)). The method is described in Appendix D.

The VI methods estimate the model with 5000 iterations of SGA, with 10 Gibbs sampling steps for the latent utilities in each SGA iteration. We take 10,000 draws from the variational posterior and predictive distribution to construct the results. The results from MCMC sampling are based on 200,000 iterations, from which the first 100,000 are discarded and we use a thinning value of 10. This results in 10,000 draws from the posterior and predictive distribution. All methods use the prior specification as discussed in Section 3.
In a multivariate multinomial probit model with uncorrelated choices, \( p \geq K \) allows for different covariance structures within each choice. Setting the number of factors \( p < J \) reduces the number of covariance parameters to be estimated, and hence reduces parameter uncertainty and computation time. Therefore, we set \( p = K \).

### 4.2 Results with 10,000 observations

#### 4.2.1 Convergence

First, we assess convergence of SGA in our VI methods. The ELBO in (15), which is typically used as convergence measure in VI, is not available in closed-form. The hit-rate defined in (21) with \( \hat{Y}_i \) the mode of \( \hat{p}_{\lambda[s]}(Y_i|X_i) \) in iteration \([s]\) can be used instead. To reduce the computational costs, we evaluate the conditional hit-rate for a fixed random subsample of 500 observations using 200 draws from \( \hat{p}_{\lambda[s]}(Y_i|X_i) \), in each tenth iteration.

Figure 1 shows the conditional hit-rate for choice 1 and 2, in VI with no subsampling and VI with 1% subsampling by a gold and black line, respectively. The figure indicates that 5000 iterations is sufficient for convergence. The conditional hit-rates remain wiggly because they are constructed using an estimate for \( \hat{p}_{\lambda[s]}(Y_i|X_i) \) that is based on a \( \lambda^{[s]} \) that is updated using an estimate for the gradient. Therefore, the final estimate for the variational parameter \( \lambda \) is constructed as the average over the \( \lambda^{[s]} \) in the final 100 iterations. We find that VI with no subsampling converges in less iterations than VI(1%). However, VI(1%) has a faster convergence, as an average iteration takes 0.018 seconds, compared to 0.562 seconds per iteration in VI.

#### 4.2.2 Parameter estimates

Second, we assess the accuracy of the parameter estimates. Table 1 shows that the VI approach provides parameter estimates of similar accuracy as MCMC, in less than 14% of the estimation time. VI with subsampling further decreases the estimation time at the costs of a loss of accuracy. The mean absolute error measures monotonically increase in the subsample size in VI. Fixing the covariance matrix to the identity matrix results in inaccurate estimates for the variances and correlations, but also for the coefficients.

Although VI only takes a fraction of the computational costs of MCMC, it still produces accurate parameter estimates. Figure 2 compares the true parameter values of \( \beta_0 \) and \( \Sigma_0 \) against the corresponding posterior mean estimates of VI and MCMC. The black and yellow circles correspond to VI and MCMC, respectively. The closer the circles lie to the 45 degree diagonal line, the closer the posterior means are to the true parameter values. Both the VI and MCMC estimates are scattered around the 45 degree line. Appendix E shows that also the VI(1%) estimates are closely located to the 45 degree line, with only a slight increase in the deviations from this line relative to VI.
This figure shows the conditional hit-rate for choice 1 and 2 in each tenth VI iteration, for VI with no subsampling and VI with 1% subsampling by a gold and black line, respectively.

Table 1: Error measures in the numerical experiment with 10,000 observations

|                  | Coefficients       | Variances          | Correlations       | Time  |
|------------------|--------------------|--------------------|--------------------|-------|
|                  | RMSE  | MAE   | RMSE  | MAE   | RMSE  | MAE   |       |
| **VI(1%)**       | 0.089 | 0.077 | 0.277 | 0.241 | 0.227 | 0.178 | 0.024 |
| **VI(10%)**      | 0.101 | 0.076 | 0.287 | 0.232 | 0.221 | 0.169 | 0.107 |
| **VI**           | 0.083 | 0.068 | 0.274 | 0.205 | 0.214 | 0.164 | 0.780 |
| **VI(I)**        | 0.166 | 0.125 | 0.333 | 0.270 | 0.290 | 0.238 | 0.434 |
| **MCMC**         | 0.075 | 0.056 | 0.301 | 0.214 | 0.215 | 0.166 | 5.794 |

This table reports the root mean squared error (RMSE) and mean absolute error (MAE) of the posterior mean estimates relative to the true parameter values. The panels for the coefficients, variances and correlations correspond to the error measures associated to the vector of coefficients $\beta_0$, the diagonal elements of $\Sigma_0$, and the implied correlations in $\Sigma_0$, respectively. The final column shows the computation time in hours. The rows denote the different estimation methods considered.

### 4.2.3 Predictive accuracy

Third, we examine the predictive accuracy of our VI approach. The log-score can be used to assess the impact of subsampling in VI on the estimated model fit. Figure 3 shows the estimation time and in- and out-of-sample log-score across choices corresponding to VI with different subsampling sizes, VI with no subsampling, and MCMC. The log-score increases in the size of the subsample, with a big increase corresponding to subsampling with 1% to 10%. After 10%, an increase in the
Figure 2: Posterior mean parameters in numerical experiment with 10,000 observations

This figure presents the estimated posterior means from MCMC (yellow circles) and VI (black circles), for the coefficients $\beta$ in Panel (a), and the variances and correlations of the latent utilities in $\Sigma$ in Panel (b) and (c), respectively.

Figure 3: Log-scores and estimation time in numerical experiment with 10,000 observations

This figure shows the log-score as defined in (20) averaged over choices 1 and 2 against the estimation time, for VI with subsampling, VI, and MCMC. The first panel shows the in-sample log-score, and the second panel the out-of-sample log-score. The log-score of the oracle is computed using the true parameter values.

Subsample results in small gains in the log-scores.

Figure 3 shows that the impact of subsampling in VI on the estimated model fit is small compared to the gains in computational efficiency. For instance, VI(10%) has an in- and out-of-sample log-score close to VI and MCMC. The differences between these three methods are small relative to the difference between MCMC and the log-score of the oracle: the log-score computed using the true parameter values. This is a striking result, as VI(10%) is estimated in 7 minutes,
Table 2: Log-score and hit-rate in numerical experiment with 10,000 observations

| Sample | Metric | VI(1%) | VI(10%) | VI | VI(I) | MCMC | naïve | oracle |
|--------|--------|--------|---------|----|-------|------|-------|--------|
| in     | log-score | -2.099 | -2.097  | -2.097 | -2.103 | -2.096 | -2.314 | -2.090 |
| in     | hit-rate   | 0.243  | 0.244   | 0.243  | 0.241  | 0.244  | 0.137  | 0.245  |
| out    | log-score  | -1.986 | -1.981  | -1.980 | -1.987 | -1.979 | -2.341 | -1.966 |
| out    | hit-rate   | 0.294  | 0.296   | 0.294  | 0.289  | 0.293  | 0.128  | 0.297  |

| Sample | Metric | VI(1%) | VI(10%) | VI | VI(I) | MCMC | naïve | oracle |
|--------|--------|--------|---------|----|-------|------|-------|--------|
| in     | log-score | -1.917 | -1.913  | -1.912 | -1.926 | -1.911 | -2.336 | -1.908 |
| in     | hit-rate   | 0.311  | 0.311   | 0.310  | 0.305  | 0.312  | 0.159  | 0.311  |
| out    | log-score  | -1.707 | -1.708  | -1.705 | -1.718 | -1.704 | -2.367 | -1.694 |
| out    | hit-rate   | 0.391  | 0.395   | 0.395  | 0.387  | 0.392  | 0.147  | 0.391  |

This table shows the in- and out-of-sample log-scores and hit-rates, defined in respectively (21) and (20). The panels correspond to the first and second choice. Predictive densities are estimated with VI with subsampling, VI, VI with an identity covariance matrix, MCMC, a naive method in which the forecast equals the most frequently observed category in the data, and the oracle that uses the true parameter values.

while VI takes 47 minutes, and MCMC almost 6 hours.

Table 2 provides a more detailed overview on the estimated model fit. The upper panel shows the in- and out-of-sample log-score and hit-rate for the first choice, and the lower panel for the second choice. These measures are higher in the second choice, indicating a stronger signal. MCMC performs better than VI on the log-scores, but does not always outperform the hit-rates of the different VI approaches. For instance, VI(10%) produces out-of-sample hit-rates that are higher than the hit-rates of MCMC in both choices. Within both choices, the performance measures of VI slowly decline with subsampling.

VI in the MVMNP model can also be compared to VI in a choice model with an identity covariance matrix VI(I). We find that VI(I) is outperformed on all measures by VI. In general, all models perform better than the naive method on all metrics: the naive methods sets the forecast equal the most frequently observed category in the data. The oracle, that uses the true parameter values to construct a forecast, corresponds to the highest log-scores, but does not always attain the highest hit-rates.

4.3 Results with one million observations

The second experiment illustrates that VI makes the estimation of the MVMNP model computationally feasible on big data sets. Figure 3 shows that MCMC takes almost six hours with 10,000 observations. In practice, choice data sets may have much larger samples as we illustrate in Section 5. With one million observations, MCMC takes around eighteen days. These computational costs make MCMC impractical in many choice applications. VI takes almost three days, which is
Table 3: Error measures in large numerical experiments

| N         | Method     | Coefficients |         |         |         |         |         |
|-----------|------------|--------------|---------|---------|---------|---------|---------|
|           |            | RMSE  MAE    | RMSE  MAE | RMSE  MAE | Time    |         |
| 10,000    | VI         | 0.083  0.068 | 0.274  0.205 | 0.214  0.164 | 46.800  |         |
| 1,000,000 | VI(1%)     | 0.073  0.054 | 0.231  0.164 | 0.195  0.150 | 48.480  |         |

This table reports the root mean squared error (RMSE) and mean absolute error (MAE) of the posterior mean estimates relative to the true parameter values. The panels for the coefficients, variances and correlations correspond to the error measures associated to the vector of coefficients $\beta_0$, the diagonal elements of $\Sigma_0$, and the implied correlations in $\Sigma_0$, respectively. The final column shows the computation time in minutes. The rows denote the number of observations and estimation method.

Table 4: Out-of-sample log-score and hit-rate in large numerical experiments

| N        | Choice 1 | Choice 2 | Average | Choice 1 | Choice 2 | Average |
|----------|----------|----------|---------|----------|----------|---------|
| 10,000   | -1.980   | -1.705   | -1.843  | 0.294    | 0.395    | 0.344   |
| 1,000,000| -1.975   | -1.701   | -1.838  | 0.296    | 0.390    | 0.343   |

This table shows the out-of-sample log-scores and hit-rates, defined in respectively (21) and (20). The parameters are estimated with VI on $N = 10,000$ observations from the data generating process and VI with 1% subsampling on $N = 1,000,000$. The predictive densities are estimated on the same out-of-sample of 10,000 observations for all three methods.

still a substantial computational cost.

Table 3 shows that VI with subsampling is scalable to a data set with one million observations. With 1% sampling the estimation only takes 48.480 minutes. The estimates have a higher accuracy than VI with 10,000 observations. The computation time of VI(1%) with one million observations is similar to VI with 10,000 observations, which demonstrates that using more observations in a similar time budget can increase accuracy.

Table 4 shows the log-scores and hit-rates of VI estimated on 10,000 observations and VI(1%) on 1,000,000 observations, on the same out-of-sample observations. The average log-score across choices increases when more than 10,000 observations are used for estimation. The average log-score of VI(1%) on 1,000,000 observations (-1.838) is even higher than the log-score of MCMC on 10,000 observations (-1.8414) in Figure 3.

5 Empirical application

To illustrate our VI method with real data, we fit a multinomial probit model to two consumer choice data sets with different dimensions. First, Section 5.1 employs a commonly used traditional
data set on laundry detergent brand purchases with a few thousand observations. Second, Section 5.2 uses a modern data set on pasta brand purchases with more than one million observations. The implementation and prior settings of the proposed methods are discussed in Section 4.1. We randomly allocate 80% of the observations for estimation of the model, and the remaining 20% are employed for out-of-sample evaluation.

5.1 Small data set with laundry detergent purchases

This section uses a small choice data set to illustrate on real data that VI is several times faster than MCMC, yet produces similar parameter estimates and predictive accuracy. The data contains 2657 purchases of six brands of laundry detergents and the log price per ounce of each brand. The data set is described in detail by Chintagunta and Prasad (1998) and available in Imai and Van Dyk (2005b). We follow Imai and Van Dyk (2005a), Burgette et al. (2021), and Loaiza-Maya and Nibbering (2021) by fitting multinomial probit models with an intercept and the log price for each brand.

Figure 4 shows that the posterior means for the parameters of VI and MCMC are similar. The yellow and black circles correspond to VI and VI(10%), respectively. The closer the circles lie to the 45 degree diagonal line, the closer the posterior means of VI are to the posterior means of MCMC. For the coefficients in panel (a), VI and VI(10%) produce posterior means that are almost identical to MCMC. The differences are small for the variances in panel (b), and a bit more pronounced for the correlations in panel (c).

The posterior distributions of the price coefficient and the posterior means of the correlation matrix further illustrate that the differences between VI and MCMC are small. Figure 5 shows that the posterior density of the coefficient of log prices mostly overlap between VI and MCMC. Figure 6 shows that the posterior mean of the correlation matrix of the latent utilities have a
This figure shows the posterior densities of the coefficient of the log prices of six laundry detergent brands of VI with a solid line, and of MCMC with a dashed line.

Figure 6: Posterior mean of the correlation matrix in laundry detergent application

This figure shows the posterior mean estimates of the elements of the correlation matrix of the latent utilities of six laundry detergent brands, with ‘Tide’ as the base category, for VI in Panel (a) and for MCMC in Panel (b).

similar pattern across the two estimation methods.

VI also attains similar predictive accuracy to MCMC. Table 5 shows that the in- and out-of-sample log-scores of VI and MCMC are almost identical. The log-scores slowly decrease in the order of subsampling in VI. The hit rates do not seem to be very sensitive to the approximations by VI, or in the VI with subsampling. Both the log-scores and hit-rates of VI(10%) are relatively close to MCMC, especially when we consider the difference in these metrics between MCMC and the naive forecasting method, in which the forecast equals the most frequently observed category in the data. The fact that VI with an identity covariance matrix results in lower log-scores than VI with a full covariance matrix, indicates that the correlations across the latent utilities matter in this application.

Although the differences in parameter estimates and predictive accuracy are minimal, VI is at least more than eight times faster than MCMC. The final row of Table 5 shows that MCMC
This table shows the in- and out-of-sample log-scores and hit-rates, defined in respectively \cite{20} and \cite{21}. The final row shows the estimation time in seconds. Predictive densities are estimated with VI with subsampling, VI, VI with an identity covariance matrix, MCMC, and a naive method in which the forecast equals the most frequently observed category in the data.

| Sample | Metric   | VI(1%) | VI(10%) | VI   | VI-I   | MCMC | Naive |
|--------|----------|--------|---------|------|--------|------|-------|
| in     | log-score| -1.345 | -1.336  | -1.332| -1.352 | -1.332| -1.639|
| in     | hit-rate | 0.490  | 0.497   | 0.501 | 0.505  | 0.494 | 0.270 |
| out    | log-score| -1.388 | -1.385  | -1.385| -1.407 | -1.383| -1.657|
| out    | hit-rate | 0.507  | 0.505   | 0.503 | 0.501  | 0.499 | 0.239 |

| time (seconds) | 15.371 | 23.413 | 80.719 | 77.262 | 720.304 |

This section shows that our approach can be scaled to real data with many observations, in contrast to existing MCMC estimation in multinomial probit models.

We use more than one million purchases from ten pasta brands in a consumer choice data set made available by Dunnhumby\cite{1} as “Carbo-Loading: A Relational Database”. From this data set, we select the purchases of pasta brands, excluding the private labels, that do not involve coupons. Since the brands with a small purchase volume are of less interest to a marketing manager, we focus on the 96.782% of purchases that corresponds to the ten top-selling pasta brands. The final sample contains 1,070,436 observations and the purchase frequencies vary from 6,280 to 316,018.

We define the log price per ounce of each brand in the same way as, for instance, Allenby and Rossi \cite{1991} and Loaiza-Mayo and Nibbering \cite{2021}. The Dunnhumby data set only contains the amount of dollar spent on a product at purchase dates. We impute the prices for brands that are not sold on a certain purchase date by taking the mean of the observed prices of a specific brand on the nearest date in the same week. In cases where there is no purchase record in the same week, we take the most recent observed price.

We consider the same MNP models as with the small data set on laundry detergent purchases, also including an intercept and the log price for each brand. As we show below, the proposed VI methods estimate the MNP parameters with the large pasta brand data within a reasonable amount of time. However, MCMC would take approximately ninety hours, which is impractical in most multinomial probit applications and especially in the retail setting we study here. Therefore we do not include MCMC as a benchmark method.

\begin{itemize}
\item takes 720 seconds while VI only takes 81 seconds. This computation time can be further reduced by subsampling. VI(10%) takes 24 seconds and VI(1%) only 16 seconds. VI(10%) is more than thirty times faster than MCMC, while we have seen that the loss in accuracy is small.
\end{itemize}

\subsection*{5.2 Large data set with pasta purchases}

This section shows that our approach can be scaled to real data with many observations, in contrast to existing MCMC estimation in multinomial probit models.

We use more than one million purchases from ten pasta brands in a consumer choice data set made available by Dunnhumby\cite{1} as “Carbo-Loading: A Relational Database”. From this data set, we select the purchases of pasta brands, excluding the private labels, that do not involve coupons. Since the brands with a small purchase volume are of less interest to a marketing manager, we focus on the 96.782% of purchases that corresponds to the ten top-selling pasta brands. The final sample contains 1,070,436 observations and the purchase frequencies vary from 6,280 to 316,018.

We define the log price per ounce of each brand in the same way as, for instance, Allenby and Rossi \cite{1991} and Loaiza-Mayo and Nibbering \cite{2021}. The Dunnhumby data set only contains the amount of dollar spent on a product at purchase dates. We impute the prices for brands that are not sold on a certain purchase date by taking the mean of the observed prices of a specific brand on the nearest date in the same week. In cases where there is no purchase record in the same week, we take the most recent observed price.

We consider the same MNP models as with the small data set on laundry detergent purchases, also including an intercept and the log price for each brand. As we show below, the proposed VI methods estimate the MNP parameters with the large pasta brand data within a reasonable amount of time. However, MCMC would take approximately ninety hours, which is impractical in most multinomial probit applications and especially in the retail setting we study here. Therefore we do not include MCMC as a benchmark method.

\begin{itemize}
\item takes 720 seconds while VI only takes 81 seconds. This computation time can be further reduced by subsampling. VI(10%) takes 24 seconds and VI(1%) only 16 seconds. VI(10%) is more than thirty times faster than MCMC, while we have seen that the loss in accuracy is small.
\end{itemize}

\section*{5.2 Large data set with pasta purchases}

This section shows that our approach can be scaled to real data with many observations, in contrast to existing MCMC estimation in multinomial probit models.

We use more than one million purchases from ten pasta brands in a consumer choice data set made available by Dunnhumby\cite{1} as “Carbo-Loading: A Relational Database”. From this data set, we select the purchases of pasta brands, excluding the private labels, that do not involve coupons. Since the brands with a small purchase volume are of less interest to a marketing manager, we focus on the 96.782% of purchases that corresponds to the ten top-selling pasta brands. The final sample contains 1,070,436 observations and the purchase frequencies vary from 6,280 to 316,018.

We define the log price per ounce of each brand in the same way as, for instance, Allenby and Rossi \cite{1991} and Loaiza-Mayo and Nibbering \cite{2021}. The Dunnhumby data set only contains the amount of dollar spent on a product at purchase dates. We impute the prices for brands that are not sold on a certain purchase date by taking the mean of the observed prices of a specific brand on the nearest date in the same week. In cases where there is no purchase record in the same week, we take the most recent observed price.

We consider the same MNP models as with the small data set on laundry detergent purchases, also including an intercept and the log price for each brand. As we show below, the proposed VI methods estimate the MNP parameters with the large pasta brand data within a reasonable amount of time. However, MCMC would take approximately ninety hours, which is impractical in most multinomial probit applications and especially in the retail setting we study here. Therefore we do not include MCMC as a benchmark method.

\begin{itemize}
\item takes 720 seconds while VI only takes 81 seconds. This computation time can be further reduced by subsampling. VI(10%) takes 24 seconds and VI(1%) only 16 seconds. VI(10%) is more than thirty times faster than MCMC, while we have seen that the loss in accuracy is small.
\end{itemize}
Figure 7: Posterior mean parameters in pasta application

This figure presents the estimated posterior means from VI(10%) (yellow circles) and VI(1%) (black circles) on the y-axis, and the estimated posterior means from VI on the x-axis, for the coefficients $\beta$ in Panel (a), and the variances and correlations of the latent utilities in $\Sigma$ in Panel (b) and (c), respectively.

Figure 8: Posterior density of the price coefficient in pasta application

This figure shows the posterior densities of the coefficient of the log prices of ten pasta brands of VI(10%) with a solid line, and of VI with a dashed line.

Figure 7 shows that the posterior means for the parameters of VI(10%) and VI are similar. The yellow and black circles correspond to VI(10%) and VI(1%), respectively. The closer the circles lie to the 45 degree diagonal line, the closer the posterior means of VI with subsampling are to the posterior means of VI without subsampling. Although a subsample of 10% does not affect the posterior means much, VI with a 1% subsample produces different estimates, especially for the correlations in Panel (c). Figure 8 and 9 show that also in the posterior distribution of the price coefficient and the posterior means of the correlation matrix of the latent utilities there are only small differences between VI(10%) and VI.

Based on the log-scores and hit-rates, VI with 10% subsampling shows almost no loss in accuracy relative to VI without subsampling. Table 6 shows that the log-score between VI(10%) and VI is almost the same. VI only shows an advantage in the fourth decimal. Although the parameter estimates are different, the difference in the log-score between VI(1%) and VI are also small, relative to the difference in this metric between VI and the naive method. This also follows
This figure shows the posterior mean estimates of the elements of the correlation matrix of the latent utilities of ten pasta brands, with ‘Barilla’ as the base category, for VI(10%) in Panel (a) and for VI in Panel (b).

Table 6: Log-score and hit-rate for pasta application

| Sample | Metric      | VI(1%)  | VI(10%) | VI    | VI-I  | Naive |
|--------|-------------|---------|---------|-------|-------|-------|
| in     | log-score   | -1.777  | -1.773  | -1.773| -1.786| -1.845|
| in     | hit-rate    | 0.397   | 0.398   | 0.398 | 0.379 | 0.295 |
| out    | log-score   | -1.777  | -1.774  | -1.774| -1.786| -1.842|
| out    | hit-rate    | 0.396   | 0.397   | 0.397 | 0.377 | 0.296 |

|                 | time (hours) | 0.175   | 1.498   | 15.253| 11.009|

This table shows the in- and out-of-sample log-scores and hit-rates, defined in respectively \(20\) and \(21\). The final row shows the estimation time in seconds. Predictive densities are estimated with VI with subsampling, VI, VI with an identity covariance matrix, and a naive method in which the forecast equals the most frequently observed category in the data.

from the the hit-rates of VI, VI(10%) and VI(1%), which are all similar. We also find that, irrespective of the subsampling size, VI outperforms VI-I, which emphasises the importance of taking correlations into account.

Table 6 also shows that VI makes the estimation of multinomial probit models feasible on real choice data sets with many observations. The final row shows that VI takes more than fifteen hours. This is a fraction of the computational costs of current MCMC estimation methods for these models, which would take ninety hours in this application. VI can even further reduce the computation time with subsampling, with VI(10%) only taking less than 1.5 hours, and VI(1%) estimates the model within eleven minutes.
6 Conclusion

Multinomial probit models are widely used for analyzing choice behavior. The main benefit of the model is the specification of the covariance matrix of the latent utilities. To accurately estimate the covariance parameters from a single categorical dependent variable, a large number of observations is required. Choice data sets with many observations are nowadays widespread available. For instance, scanner data as used in the empirical application in this paper have records of millions of transactions. However, MCMC methods that are currently used for parameter estimation are computationally costly.

This paper proposes a variational inference method that employs the conditional posterior of the latent utilities as a part of the variational family. This allows for accurate approximations to the exact posterior. The method is faster than MCMC with moderately sized data sets, and is scalable to large-scale data in which MCMC estimation is infeasible.

Numerical experiments and an empirical application to a laundry detergent choice set demonstrate that our approach produces accurate approximating densities to the MCMC exact posterior densities, while only requiring a small fraction of the MCMC computation time. The computational cost for our approach can be further reduced by considering subsampling methods inside the stochastic gradient ascent algorithm, with small impact in its predictive accuracy relative to MCMC.

The new method substantially improves the practical applicability of the multinomial probit model to modern choice data sets. We illustrate the potential of the new approach in large samples by applying it to a pasta choice data set that consists of more than one million observations, in which MCMC is not feasible. Our proposed method can be implemented for this data set in less than sixteen hours, and if subsampling is considered in less than eleven minutes.

References

Albert, J. H. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American statistical Association*, 88(422):669–679.

Allenby, G. M. and Rossi, P. E. (1991). Quality perceptions and asymmetric switching between brands. *Marketing science*, 10(3):185–204.

Bunch, D. S. (1991). Estimability in the multinomial probit model. *Transportation Research Part B: Methodological*, 25(1):1–12.

Burgette, L. F. and Nordheim, E. V. (2012). The trace restriction: An alternative identification strategy for the Bayesian multinomial probit model. *Journal of Business & Economic Statistics*, 30(3):404–410.
Burgette, L. F., Puelz, D., and Hahn, P. R. (2021). A symmetric prior for multinomial probit models. *Bayesian Analysis*, pages 1–18.

Chib, S., Greenberg, E., Chen, Y., et al. (1998). MCMC methods for fitting and comparing multinomial response models. *Economics Working Paper Archive, Econometrics*, 9802001.

Chintagunta, P. K. and Prasad, A. R. (1998). An empirical investigation of the “dynamic McFadden” model of purchase timing and brand choice: Implications for market structure. *Journal of Business & Economic Statistics*, 16(1):2–12.

Fasano, A. and Durante, D. (2022). A class of conjugate priors for multinomial probit models which includes the multivariate normal one. *Journal of Machine Learning Research*, 23(30):1–26.

Girolami, M. and Rogers, S. (2006). Variational Bayesian multinomial probit regression with Gaussian process priors. *Neural Computation*, 18(8):1790–1817.

Imai, K. and Van Dyk, D. A. (2005a). A Bayesian analysis of the multinomial probit model using marginal data augmentation. *Journal of econometrics*, 124(2):311–334.

Imai, K. and Van Dyk, D. A. (2005b). MNP: R package for fitting the multinomial probit model. *Journal of Statistical Software*, 14(3):1–32.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*.

Loaiza-Maya, R. and Nibbering, D. (2021). Scalable Bayesian estimation in the multinomial probit model. *Journal of Business & Economic Statistics*, pages 1–13.

Loaiza-Maya, R., Smith, M. S., Nott, D. J., and Danaher, P. J. (2021). Fast and accurate variational inference for models with many latent variables. *Journal of Econometrics*, pages 1–24.

McCulloch, R. and Rossi, P. E. (1994). An exact likelihood analysis of the multinomial probit model. *Journal of Econometrics*, 64(1-2):207–240.

McCulloch, R. E., Polson, N. G., and Rossi, P. E. (2000). A Bayesian analysis of the multinomial probit model with fully identified parameters. *Journal of Econometrics*, 99(1):173–193.

Mishkin, E. (2021). Gender and sibling dynamics in the intergenerational transmission of entrepreneurship. *Management Science*, 67(10):6116–6135.

Miyazaki, K., Hoshino, T., and Böckenholt, U. (2021). Dynamic two stage modeling for category-level and brand-level purchases using potential outcome approach with Bayes inference. *Journal of Business & Economic Statistics*, 39(3):622–635.
Ong, V. M.-H., Nott, D. J., and Smith, M. S. (2018). Gaussian variational approximation with a factor covariance structure. *Journal of Computational and Graphical Statistics*, 27(3):465–478.

Ormerod, J. T. and Wand, M. P. (2010). Explaining variational approximations. *The American Statistician*, 64(2):140–153.

Paleti, R. (2018). Generalized multinomial probit model: Accommodating constrained random parameters. *Transportation Research Part B: Methodological*, 118:248–262.

Richard Hahn, P., Carvalho, C. M., and Scott, J. G. (2012). A sparse factor analytic probit model for congressional voting patterns. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 61(4):619–635.

Smith, M. S., Loaiza-Mayaa, R., and Nott, D. J. (2020). High-dimensional copula variational approximation through transformation. *Journal of Computational and Graphical Statistics*, 29(4):729–743.

Talhouk, A., Doucet, A., and Murphy, K. (2012). Efficient Bayesian inference for multivariate probit models with sparse inverse correlation matrices. *Journal of Computational and Graphical Statistics*, 21(3):739–757.

Westling, T. and McCormick, T. (2019). Beyond prediction: A framework for inference with variational approximations in mixture models. *Journal of Computational and Graphical Statistics*, 28(4):778–789.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zhang, X., Boscardin, W. J., and Belin, T. R. (2006). Sampling correlation matrices in Bayesian models with correlated latent variables. *Journal of Computational and Graphical Statistics*, 15(4):880–896.

Zhang, X., Boscardin, W. J., and Belin, T. R. (2008). Bayesian analysis of multivariate nominal measures using multivariate multinomial probit models. *Computational statistics & data analysis*, 52(7):3697–3708.
A Prior specification

The prior distribution is defined as \( p(\theta) = p(\beta) \prod_{k=1}^{K} p(\kappa_k), \) with \( \beta \sim N(0, \frac{1}{10} I_r), \) \( p(\kappa_k) = \prod_{l=1}^{n_k-1} p(\kappa_{kl}), \) and \( p(\kappa_{kl}) = \phi_1 \left\{ t_{\hat{\eta}_{kl}} \left[ \frac{g(\kappa_{kl}) - \hat{\mu}_{kl}}{\hat{\tau}_{kl}} \right] \right\} \psi_{kl} \left[ \frac{g(\kappa_{kl}) - \hat{\mu}_{kl}}{\hat{\tau}_{kl}} \right] \frac{1}{\hat{\tau}_{kl}} G^2(\kappa_{kl}). \) The hyperparameters \( \hat{\mu}_{kl}, \hat{\tau}_{kl}, \) and \( \hat{\eta}_{kl} \) are calibrated for each choice \( k \) at a time using Algorithm 1 in Loaiza-Maya and Nibbering (2021).

The algorithm calibrates the prior for \( \kappa_k \) using the following prior on \( \psi: \)

\[
\psi_k = \frac{\sqrt{J_k}}{\| \psi_k \|} \hat{\psi}_k, \tag{22}
\]

\[
p(\hat{\psi}_k | \theta) = \prod_{j=1}^{J} \left[ p(d_{kj} | \nu, s) \prod_{l=1}^{J} p(B_{kl}) \sigma_B^2 \right], \tag{23}
\]

\[
\hat{B}_{kl} \sigma_B^2 \sim N(\mu_B, \sigma_B^2), \text{ if } j \neq l, \tag{24}
\]

\[
\hat{B}_{kij} \sigma_B^2 \sim N(\mu_B, \sigma_B^2) I(\hat{B}_{kij} > 0), \tag{25}
\]

\[
d_{kj}^2 | \nu, s \sim \text{Inverse-Gamma}(\nu, s), \tag{26}
\]

where \( \nu \) and \( s \) denote the shape and rate parameters of the Inverse-Gamma distribution.

The angles \( \kappa_{kl} \) are transformed to the real line as

\[
\xi_{kl} = \begin{cases} 
G(\kappa_{kl}) = \Phi^{-1} \left( \frac{2\mu}{\pi} \right) & \text{if } l < n_k - 1, \\
G(\kappa_{kl}) = \Phi^{-1} \left( \frac{2\mu}{\pi} \right) & \text{if } l = n_k - 1. 
\end{cases} \tag{27}
\]

After transforming, we use the Jacobian of the transformation to obtain the prior density

\[
p(\xi_{kl}) = \phi_1 \left\{ t_{\hat{\eta}_{kl}} \left( \frac{\hat{\xi}_{kl} - \hat{\mu}_{kl}}{\hat{\tau}_{kl}} \right) \right\} \psi_{kl} \left( \frac{\hat{\xi}_{kl} - \hat{\mu}_{kl}}{\hat{\tau}_{kl}} \right) \frac{1}{\hat{\tau}_{kl}}, \tag{28}
\]

and the full prior density for \( \theta = (\beta^\top, \xi^\top)^\top \) can be written as

\[
p(\theta) = p(\beta) \prod_{k=1}^{K} \prod_{l=1}^{n_k-1} p(\xi_{kl}). \tag{29}
\]

B Monte Carlo Markov Chain sampling scheme

Define \( L_k = \sum_{l=1}^{k} J_l. \) Define the \( L_K \)-dimensional vectors \( Y_i = (Y_{ik1}^\top, \ldots, Y_{ikK}^\top)^\top \) and \( Z_i = (Z_{i1}^\top, \ldots, Z_{iK}^\top)^\top, \) and the \( L_K \times q \) matrix \( X_i = (X_{i1}^\top, \ldots, X_{iK}^\top)^\top. \) Define \( Y = (Y_1^\top, \ldots, Y_N^\top)^\top, \) \( Z = (Z_1^\top, \ldots, Z_N^\top)^\top, \) and \( X = (X_1^\top, \ldots, X_N^\top)^\top. \) Samplings steps for \( \beta \) and \( Z \) are standard and also discussed in, for instance, Zhang et al. (2008).
B.1 Conditional posterior $\beta$

Generate from $\beta | Z, \Sigma, X$: The coefficients $\beta$ are generated from

$$\beta | Z, \Sigma, X \sim \mathcal{N}(\bar{b}, \bar{B}^{-1}),$$

with $\bar{B} = X^*^T X^* + B$ and $\bar{b} = \bar{B}^{-1} X^*^T Z^*$, where $X^* = (X_1^T C, \ldots, X_N^T C)^T$ and $Z^* = (Z_1^T C, \ldots, Z_N^T C)^T$, with $\Sigma^{-1} = CC^\top$.

B.2 Conditional posterior $Z$

Generate from $Z | \beta, \Sigma, Y, X$. To generate from the latent utilities we employ the truncated normal distributions,

$$Z_{ikj} \sim N^+_{\max(Z_{ikj}^{(j)}, 0)}(\bar{\mu}_{ikj}, \bar{\Sigma}_{ikj}), \text{ if } Y_{ikj} = j,$$

$$Z_{ikj} \sim N^-_{\max(Z_{ikj}^{(j)}, 0)}(\bar{\mu}_{ikj}, \bar{\Sigma}_{ikj}), \text{ if } Y_{ikj} \neq j,$$

where $N^+_{\mu, \sigma^2}$ and $N^-_{\mu, \sigma^2}$ represent a normal distribution with mean $\mu$ and variance $\sigma^2$ truncated from below or above by $a$, respectively, and $Z^{(j)}_{ik} = (Z_{ik1}, \ldots, Z_{ikj-1}, Z_{ikj+1}, \ldots, Z_{ikL})^T$. The conditional mean and variance of $Z_{ikj}$ given $Z^{(k)}_i = (Z_{i1}^T, Z_{i2}^T, Z_{i3}^T, Z_{i4}^T, Z_{i5}^T)^T$, are defined as

$$\bar{\mu}_{ikj} = X_{ikj}^\top \beta + \Sigma_{kj(kj)}^{-1}(Z^{(k)}_{ik} - X_{ikj}^\top \beta),$$

$$\bar{\Sigma}_{ikj} = \Sigma_{kkj}^{-1} - \Sigma_{kkj}^{-1}(\Sigma_{kj(kj)}^{-1} - \Sigma_{(kj)(kj)}) \Sigma_{kj(kj)}^{-1},$$

where $X_{ikj}^{(k)}$ is defined in the same way as $Z^{(k)}_{ik}$, $\Sigma_{kj(kj)}$ is the element in row and column number $L_{k-1} + j$ of $\Sigma$, $\Sigma_{kj(kj)}$ is row (column) number $L_{k-1} + j$ of $\Sigma$ without column (row) number $L_{k-1} + j$, and $\Sigma_{(kj)(kj)}$ equals $\Sigma$ after removing row and column number $L_{k-1} + j$.

B.3 Conditional posterior $\kappa$

Sampling of the parameters $\kappa$ is obtained via blocked random walk Metropolis-Hastings steps. At the start of each iteration, randomly allocate the elements of $\kappa$ into $G$ parameter blocks, $\kappa_{b_1}, \ldots, \kappa_{b_G}$, of five elements each. Note that in this appendix each index $i$ in $\kappa_i$ refers to a block of random elements of $\kappa$ and not to the choice-specific vector of angles as in Section 2.3. For $g = 1, \ldots, G$, generate a draw $\kappa_{b_g}^{\text{new}}$ from the proposal density,

$$q(\kappa_{b_g} | \kappa_{b_g}^{\text{old}}) = \prod_{l=1}^{5} \Phi_1(\kappa_{b_g,l}^{\text{old}}, \kappa_{b_g,l}^{\text{old}}, \sigma_g^2) - \Phi_1(\kappa_{b_g,l}^{\text{new}}, \kappa_{b_g,l}^{\text{new}}, \sigma_g^2),$$

(35)
where $\kappa_{b_l}$ is the $l$th element of $\kappa_{b}$. The constants low and up denote the lower and upper bounds of $\kappa_{b_l}$. The proposal parameters $\sigma_{2}^2$ are set adaptively to target acceptance rates between 15% and 30%. We accept $\kappa_{b_l}^{\text{new}}$ with probability

$$\alpha = \min \left( 1, \frac{p(\kappa_{b_l}^{\text{new}} | z, X, \{ \theta \setminus \kappa_{b_l} \}) q(\kappa_{b_l}^{\text{old}} | \kappa_{b_l}^{\text{new}})}{p(\kappa_{b_l}^{\text{old}} | z, X, \{ \theta \setminus \kappa_{b_l} \}) q(\kappa_{b_l}^{\text{new}} | \kappa_{b_l}^{\text{old}})} \right),$$

(36)

where

$$p(\kappa_{b_l} | z, X, \{ \theta \setminus \kappa_{b_l} \}) \propto p(\kappa_{b_l}) p(z | X, \theta).$$

(37)

The expression $\{ \theta \setminus \kappa_{b_l} \}$ denotes the subtraction of the subset $\kappa_{b_l}$ from $\theta$.

C VI in the MVMNP model

VI in the MVMNP model requires an unbiased estimate of the gradient

$$\nabla_\theta \log g(\theta, z) = \left( \nabla_\beta \log g(\theta, z)^T, \nabla_\xi \log g(\theta, z)^T \right)^T,$$

where $\theta = (\beta^T, \xi^T)^T$, $\xi = (\xi_1^T, \ldots, \xi_K^T)^\top$, $\xi_k = (\xi_{k,1}, \ldots, \xi_{k,n_k-1})^T$, and $\xi_{kl}$ is defined in [27]. The function $\log g(\theta, z)^T$ can be written as

$$\log g(\theta, z) = \log p(y | z) + \log p(\theta) + \sum_{i=1}^N \log \phi_J(z_i; X_i \beta, \Sigma(\xi)),$$

(38)

where

$$\log p(\theta) = \log p(\beta) + \sum_{k=1}^K \sum_{l=1}^{n_k-1} \log p(\xi_{kl}),$$

(39)

and $\Sigma(\xi)$ reflects the fact that the covariance matrix now depends on $\xi$.

The estimate can be constructed on a set $A \subseteq \{1, \ldots, N\}$ of $M$ indexes sampled at random and without replacement:

$$\nabla_\beta \log g(\theta, z_A) = \nabla_\beta \log p(\theta) + \frac{N}{M} \sum_{i \in A} \nabla_\beta \log \phi_J(z_i; X_i \beta, \Sigma(\xi)),$$

(40)

$$\nabla_\xi \log g(\theta, z_A) = \nabla_\xi \log p(\theta) + \frac{N}{M} \sum_{i \in A} \nabla_\xi \log \phi_J(z_i; X_i \beta, \Sigma(\xi)),$$

(41)

which boils down to the exact evaluation of $\nabla_\theta \log g(\theta, z)$ when $N = M$ and there is no subsampling. We show how to evaluate the terms $\nabla_\beta \log p(\theta)$, $\nabla_\xi \log p(\theta)$, $\nabla_\beta \log \phi_J(z_i; X_i \beta, \Sigma(\xi))$ and $\nabla_\xi \log \phi_J(z_i; X_i \beta, \Sigma(\xi))$ below. Throughout we use $\eta_i = z_i - X_i \beta$. 

27
\( C.1 \) Computing \( \nabla_\beta \log p(\theta) \)

\[
\nabla_\beta \log p(\theta) = -10\beta. \tag{42}
\]

\( C.2 \) Computing \( \nabla_\xi \log p(\theta) \)

To compute this gradient we know that

\[
\nabla_\xi \log p(\theta) = \left( \frac{\partial \log p(\xi_{k,1})}{\partial \xi_{k,1}}, \ldots, \frac{\partial \log p(\xi_{k,n_k-1})}{\partial \xi_{k,n_k-1}} \right)^\top,
\]

and

\[
\frac{\partial \log p(\xi_{kl})}{\partial \xi_{kl}} = -\frac{1}{\tilde{\tau}_{kl}} t_{\tilde{\eta}_{kl}} \left[ \frac{\xi_{kl} - \hat{\mu}_{kl}}{\tilde{\tau}_{kl}} \right]^T t'_{\tilde{\eta}_{kl}} \left[ \frac{\xi_{kl} - \hat{\mu}_{kl}}{\tilde{\tau}_{kl}} \right] + \frac{1}{\tilde{\tau}_{kl}} t''_{\tilde{\eta}_{kl}} \left[ \frac{\xi_{kl} - \hat{\mu}_{kl}}{\tilde{\tau}_{kl}} \right].
\]

Closed-form expressions for \( t_{\tilde{\eta}_{kl}}(\cdot), t'_{\tilde{\eta}_{kl}}(\cdot) \) and \( t''_{\tilde{\eta}_{kl}}(\cdot) \) are provided in Table 1 in Smith et al. (2020).

\( C.3 \) Computing \( \nabla_\beta \log \phi_J(z_i; X_i \beta, \Sigma(\xi)) \)

\[
\nabla_\beta \log \phi_J(z_i; X_i \beta, \Sigma(\xi))^\top = \eta_i^\top \Sigma^{-1} X_i. \tag{43}
\]

\( C.4 \) Computing \( \nabla_\xi \log \phi_J(z_i; X_i \beta, \Sigma(\xi)) \)

We use the convention that for two generic matrices \( C_{\text{dim1} \times \text{dim2}} \) and \( L_{\text{dim3} \times \text{dim4}} \) we have that

\[
\frac{\partial C}{\partial L} = \frac{\partial \text{vec}(C)}{\partial \text{vec}(L)} = E,
\]
where $E$ is of dimension $(\text{dim1} \times \text{dim2}) \times (\text{dim3} \times \text{dim4})$. We apply the chain rule of derivatives multiple times to write

$$
\nabla_\xi \log \phi_j (z_i; X_i\beta, \Sigma(\xi))^T = \frac{\partial}{\partial \xi} \left[ -\frac{1}{2} \log(2\pi) - \frac{1}{2} \log(\det(\Sigma)) - \frac{1}{2} \eta_i^\top \Sigma^{-1} \eta_i \right]
$$

$$
= \frac{1}{2} \frac{\partial}{\partial \xi} \log(\det(\Sigma)) + \frac{1}{2} \eta_i^\top \Sigma^{-1} \eta_i
$$

$$
= -\frac{1}{2} \frac{\partial}{\partial \xi} \log(\det(\Sigma)) - \frac{1}{2} \eta_i^\top \Sigma^{-1} \eta_i
$$

$$
= \frac{1}{2} \frac{\partial}{\partial \xi} \log(\det(\Sigma)) - \frac{1}{2} \eta_i^\top \Sigma^{-1} \eta_i
$$

$$
= \frac{1}{2} \frac{\partial}{\partial \xi} \log(\det(\Sigma)) - \frac{1}{2} \eta_i^\top \Sigma^{-1} \eta_i
$$

$$
= \frac{1}{2} \frac{\partial}{\partial \xi} \log(\det(\Sigma)) - \frac{1}{2} \eta_i^\top \Sigma^{-1} \eta_i
$$

$$
= \frac{1}{2} \frac{\partial}{\partial \xi} \log(\det(\Sigma)) - \frac{1}{2} \eta_i^\top \Sigma^{-1} \eta_i
$$

where the fourth line uses that $\text{vec}(\Sigma) = (\Omega, \Omega')$. The sixth line uses that $\frac{\partial\Sigma}{\partial \kappa} = \frac{\partial\Sigma}{\partial \beta} \frac{\partial\kappa}{\partial \beta} + \frac{\partial\Sigma}{\partial \beta} \frac{\partial\kappa}{\partial \beta}$. Note that $\frac{\partial\det(\Sigma)}{\partial \xi} = \det(\Sigma)\text{vec}(\Sigma^{-1})^T$. We derive the remaining expressions below.

C.4.1 Derivation of $\frac{1}{2} (\eta_i^\top \otimes \eta_i^\top) \frac{\partial\Sigma^{-1}}{\partial \beta} \frac{\partial\beta}{\partial \kappa}$

We have $\frac{\partial\Sigma^{-1}}{\partial \beta} = - (\Sigma^{-1} \otimes \Sigma^{-1})$ and $\frac{\partial\beta}{\partial \kappa} = (I_{j^2} + K_{j,J}) (\Sigma \otimes I_j)$, where $K_{m,n}$ is the commutation matrix of an $m \times n$ matrix. The product of these two derivatives can be simplified as

$$
\frac{\partial\Sigma^{-1}}{\partial \beta} \frac{\partial\beta}{\partial \kappa} = - (I_{j^2} + K_{j,J}) (\Sigma^{-1} B \otimes \Sigma^{-1})
$$

(44)

It follows that

$$
\frac{1}{2} (\eta_i^\top \otimes \eta_i^\top) \frac{\partial\Sigma^{-1}}{\partial \beta} \frac{\partial\beta}{\partial \kappa} = - \frac{1}{2} (\eta_i^\top \otimes \eta_i^\top) (I_{j^2} + K_{j,J}) (\Sigma^{-1} B \otimes \Sigma^{-1})
$$

(45)

$$
= - \frac{1}{2} (\eta_i^\top \Sigma^{-1} B \otimes \eta_i^\top \Sigma^{-1}) - \frac{1}{2} (\eta_i^\top \Sigma^{-1} \otimes \eta_i^\top \Sigma^{-1}) K_{j,q}
$$

(46)

$$
= - (\eta_i^\top \Sigma^{-1} B \otimes \eta_i^\top \Sigma^{-1}).
$$

(47)

C.4.2 Derivation of $\frac{1}{2} (\eta_i^\top \otimes \eta_i^\top) \frac{\partial\Sigma^{-1}}{\partial d} \frac{\partial d}{\partial \kappa}$

Following identical steps as above we can show that

$$
\frac{1}{2} (\eta_i^\top \otimes \eta_i^\top) \frac{\partial\Sigma^{-1}}{\partial d} = - (\eta_i^\top \Sigma^{-1} D \otimes \eta_i^\top \Sigma^{-1}) P,
$$

(48)

where $P$ extracts the columns corresponding to the elements of $d$. 

29
C.4.3 Derivation of $\frac{\partial B}{\partial \kappa}$

Denote $C_k = [B_k|d_k]$ and define $C = [C_1^T|\ldots|C_K^T]^T$. Note that $B = CP_3$, $d = CP_4$ for $P_3 = [I_q|0_{q\times 1}]^T$ and $P_2 = (0_{1\times q}, 1)^T$. With this notation we can then write

$$\frac{\partial B}{\partial \kappa} = \frac{\partial B}{\partial C} \frac{\partial C}{\partial \kappa}.$$  

(49)

The first term can be derived from the expression

$$\text{vec} \left( B \right) = \left( P_3^T \otimes I_J \right) \text{vec} \left( C \right),$$

(50)

from which we obtain $\frac{\partial B}{\partial C} = \left( P_3^T \otimes I_J \right)$. The second term can be derived as

$$\frac{\partial C}{\partial \kappa} = K_{q+1,j} \frac{\partial C^T}{\partial \kappa}$$

where

$$\frac{\partial C^T}{\partial \kappa} = \text{blockdiag} \left( \frac{\partial C_1^T}{\partial \kappa_1}, \ldots, \frac{\partial C_K^T}{\partial \kappa_K} \right),$$

(51)

$$\frac{\partial C_k^T}{\partial \kappa_k} = K_{J_k,q+1} \frac{\partial C_k}{\partial \kappa_k},$$

and the elements of $\frac{\partial C_k}{\partial \kappa_k}$ equal $\left\{ \frac{\partial C_k}{\partial \kappa_k} \right\}_{l,j} = \frac{\partial \psi_{kl}(\kappa_k)}{\partial \kappa_k}$ with

$$\frac{\partial \psi_{kl}(\kappa_k)}{\partial \kappa_k} = \begin{cases} 
\sqrt{J_k} \cos(k_{k,j}) \cos(k_{kl}) \prod_{s \in \{1,\ldots,l-1\}\setminus j} \sin(k_{k,s}) & \text{if } j < l \text{ and } l < n_k, \\
-\sqrt{J_k} \prod_{s=1}^{l-1} \sin(k_{k,s}) & \text{if } j = l \text{ and } l < n_k, \\
\sqrt{J_k} \cos(k_{k,j}) \prod_{s \in \{1,\ldots,l-1\}\setminus j} \sin(k_{k,s}) & \text{if } j < l \text{ and } l = n_k, \\
0 & \text{if otherwise.} 
\end{cases}$$

(52)

C.4.4 Derivation of $\frac{\partial d}{\partial \kappa}$

Similarly, we have that

$$\frac{\partial d}{\partial \kappa} = \frac{\partial d}{\partial C} \frac{\partial C}{\partial \kappa}.$$  

(53)

We calculated the second term previously. Following the same logic as before, the first term can be computed from noting that

$$\text{vec} \left( d \right) = \left( P_4^T \otimes I_J \right) \text{vec} \left( C \right),$$

(54)

so that $\frac{\partial d}{\partial C} = \left( P_4^T \otimes I_J \right)$. 

30
**D VI with the identity covariance matrix**

The variational approach proposed in this paper can also be applied to estimation of the MVMNP model with identity covariance matrix. The main difference is that this model does not have a vector of angles \( \kappa \). When \( \Sigma \) is fixed at the identity matrix, VI only requires an unbiased estimate of the gradient \( \nabla_\beta \log g(\beta, z) \). The function \( \log g(\beta, z)^T \) can be written as

\[
\log g(\beta, z) = \log p(y|z) + \log p(\beta) + \sum_{i=1}^N \log \phi_J(z_i; X_i\beta, I_J).
\] (55)

The estimate can be constructed on a set \( A \subseteq \{1, \ldots, N\} \) of \( M \) indexes sampled at random and without replacement:

\[
\nabla_\beta \log g(\beta, z_A) = \nabla_\beta \log p(\beta) + \frac{N}{M} \sum_{i \in A} \nabla_\beta \log \phi_J(z_i; X_i\beta, I_J),
\] (56)

Appendix C.1 shows how to evaluate \( \nabla_\beta \log p(\beta) \), and Appendix C.3 provides an expression for \( \nabla_\beta \log \phi_J(z_i; X_i\beta, \Sigma(\xi)) \), which after replacing \( \Sigma(\xi) = I_J \) provides the required gradient.

**E Additional results numerical experiments**

Figure 10: Posterior mean parameters in numerical experiment with 10,000 observations

This figure presents the estimated posterior means from MCMC (yellow circles) and VI(1\%) (black circles), for the coefficients \( \beta \) in Panel (a), and the variances and correlations of the latent utilities in \( \Sigma \) in Panel (b) and (c), respectively.