Stochastic variational inference for large-scale discrete choice models using adaptive batch sizes

Linda Tan

National University of Singapore

Seminar at Institute for Choice UniSA (9 July 2014)

Motivation

- Mixed multinomial logit model: captures heterogeneity in preferences of decision makers through random coefficients
- Classical approach: Maximize simulated likelihood (McFadden & Train 2000)
- Bayesian approach: Markov chain Monte Carlo (MCMC) methods
  - Gibbs sampling + Metropolis-Hastings algorithm (Rossi et al. 2005)
  - Avoid convergence issues in classical approach
  - Consistency and efficiency under fewer restrictions (Train 2009)
- MCMC computations prohibitively expensive for large datasets
- Variational methods offer competitive accuracy at lower computational cost (Braun & McAuliffe 2010)

Proposed Methods

- Explore alternative variational methods that allow posterior independence assumption among random coefficients to be dropped
- Use stochastic variational inference to accelerate convergence for large datasets (data processed in minibatches)
- Novel strategy to increase minibatch sizes adaptively

Mixed multinomial logit models of discrete choice

- T_h choice events observed for each agent h, $h = 1, \dots, H$
- Agent selects from J alternatives at each choice event
- Utility agent h obtains from alternative j at tth choice event:

$$U_{htj} = x_{htj}^T \beta_h + \epsilon_{htj}$$

- x_{htj}: vector of observed variables that relate to alternative j and agent h at tth choice event
- β_h: random vector of coefficients for agent h
- ϵ_{htj}: random error term representing unobserved utility

Mixed multinomial logit model

- $y_{ht} = [y_{ht}^1, \dots, y_{ht}^J]^T$: $J \times 1$ indicator vector denoting outcome of agent h at tth choice event and $x_{ht} = [x_{ht1}, \dots, x_{htJ}]^T$.
- Assume random errors ϵ_{htj} are iid extreme value and

$$\beta_h \sim N(\zeta, \Omega) \quad \text{for} \quad h = 1, \dots, H.$$

- Choice probabilities:

$$P(y_{ht}^j = 1 | x_{ht}, \beta_h) = \frac{\exp(x_{htj}^T \beta_h)}{\sum_{j'=1}^J \exp(x_{htj'}^T \beta_h)} \;\text{ for }\; j = 1, \dots, J,$$

$$p(y_{ht} | x_{ht}, \beta_h) = \prod_{j=1}^J \left\{ \frac{\exp(x_{htj}^T \beta_h)}{\sum_{j'=1}^J \exp(x_{htj'}^T \beta_h)} \right\}^{y_{ht}^j}$$

Bayesian approach to inference

- Priors:

$$\zeta \sim N(\mu_0, \Sigma_0)$$

$$\Omega \sim IW\left(\nu + K - 1,\ 2\nu\,\mathrm{diag}(1/a)\right),\quad a = [a_1, \ldots, a_K]^T$$

$$a_k \overset{\text{iid}}{\sim} IG(1/2,\ 1/A_k^2),\quad A_k > 0 \ \text{ for } \ k = 1, \ldots, K.$$

- Hyperparameters μ_0, Σ_0, ν and $A_1, \ldots, A_K$ considered known
- Priors for Ω are marginally noninformative (Huang & Wand 2013)
- Large A_k: weakly informative Half-t distributions on standard deviation terms in Ω
- $\nu = 2$: marginal uniform distributions for correlation terms in Ω
- Examples: $\zeta \sim N(0, 10^6)$, $\nu = 2$, $A_k = 10^3$ for $k = 1, \ldots, K$

Mixed multinomial logit model

- Unknown parameters: $\theta = \{\beta, \zeta, \Omega,\ a\}$ where $\beta = [\beta_1^T, \ldots, \beta_H^T]^T$
- Global variables: ζ, Ω, a (common across all agents)
- Local variables: β_h (specific to a particular agent)
- Joint density:

$$p(y, \theta) = \left\{ \prod_{k=1}^{K} p(a_k | A_k) \right\} p(\Omega | \nu, a) p(\zeta | \mu_0, \Sigma_0) \\ \times \prod_{h=1}^{H} p(\beta_h | \zeta, \Omega) \prod_{t=1}^{T} p(y_{ht} | x_{ht}, \beta_h)$$

Introduction to variational methods

- Approximate $p(\theta|y)$ by more tractable density function $q(\theta)$
- Minimize Kullback-Leibler divergence between $q(\theta)$ and $p(\theta|y)$

$$\log p(y) = \underbrace{\int q(\theta) \log \frac{p(y,\theta)}{q(\theta)}\, d\theta}_{\text{Lower bound } (\mathcal{L})} + \underbrace{\int q(\theta) \log \frac{q(\theta)}{p(\theta|y)}\, d\theta}_{\text{Kullback-Leibler divergence} \geq 0}$$

- Maximizing $\mathcal{L} \Leftrightarrow$ minimizing Kullback-Leibler divergence

Variational Bayes (Attias, 1999)

- Assume $q(\theta) = \prod_{i=1}^{m} q_i(\theta_i)$ for $\theta = \{\theta_1, \ldots, \theta_m\}$
- Optimal densities maximizing $\mathcal{L}$ satisfy

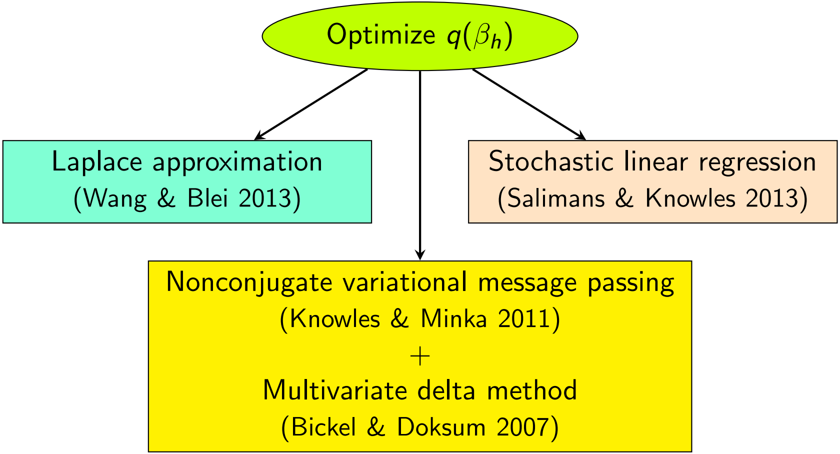
$$q_i(\theta_i) \propto \exp E_{-\theta_i}\{\log p(y, \theta)\} \quad \text{for} \quad i = 1, \ldots, m.$$

- $E_{-\theta_i}$: expectation w.r.t. $\prod_{j \neq i} q_j(\theta_j)$
- Conjugate priors:
  - optimal q_i belong to recognizable density families
  - suffice to optimize parameters of q_i

Variational Bayes for mixed multinomial logit model

- Assume
$$q(\theta) = q(\zeta)q(\Omega)q(a)\prod_{h=1}^{H} q(\beta_h)$$
- $q(\zeta)$, $q(\Omega)$ and $q(a)$: conjugate priors
- Optimal densities: $q(\zeta) = N(\mu_\zeta, \Sigma_\zeta)$, $q(\Omega) = IW(\omega, \Upsilon)$ and $q(a) = \prod_{k=1}^{K} q(a_k)$ where $q(a_k) = IG(b_k, c_k)$
- Likelihood $p(y_{ht}|x_{ht}, \beta_h)$ is nonconjugate w.r.t. prior over β_h
- Optimal $q(\beta_h)$ does not belong to any recognizable density family

Optimizing local variational parameters



Optimize $q(\beta_h)$

Laplace approximation
(Wang & Blei 2013)

Stochastic linear regression
(Salimans & Knowles 2013)

Nonconjugate variational message passing
(Knowles & Minka 2011)
$+$
Multivariate delta method
(Bickel & Doksum 2007)

Laplace approximation

- $p(\theta|y)$: intractable posterior density
- Second-order Taylor approximation to $\log p(\theta|y)$ at maximum a posterior (MAP) estimate $\hat{\theta}$

$$\log p(\theta|y) \approx \log p(\hat{\theta}|y) + \frac{1}{2}(\theta - \hat{\theta})^T H(\hat{\theta})(\theta - \hat{\theta}),$$

- $H(\hat{\theta}) = \nabla^2 \log p(\hat{\theta}|y)$.
- $\nabla \log p(\hat{\theta}|y) = 0$ since $\log p(\theta|y)$ is maximized at $\hat{\theta}$.
- Gaussian approximation:

$$p(\theta|y) \approx N(\hat{\theta}, -H(\hat{\theta})^{-1}).$$

Laplace variational inference

- Apply Laplace approximation within variational Bayes optimal density update
- Optimal $q(\beta_h)$ satisfies

$$\begin{aligned} q(\beta_h) &\propto \exp E_{-\beta_h}\left\{\sum_{t=1}^{T_h} \log p(y_{ht}|x_{ht}, \beta_h) + \log p(\beta_h|\zeta, \Omega)\right\} \\ &\propto \exp\{f(\beta_h)\} \end{aligned}$$

where

$$\begin{aligned} f(\beta_h) = \sum_{t=1}^{T_h}\left[y_{ht}^T x_{ht}\beta_h - \log\left\{\sum_{j=1}^{J} \exp\left(x_{htj}^T\beta_h\right)\right\}\right] \\ - \frac{\omega}{2}(\beta_h - \mu_\zeta)^T \Upsilon^{-1}(\beta_h - \mu_\zeta). \end{aligned}$$

Laplace variational inference

- Suppose $f(\beta_h)$ is maximized at $\hat{\beta}_h$ so that $\nabla f(\hat{\beta}_h) = 0$
- Second-order Taylor approximation of $f(\beta_h)$ at $\hat{\beta}_h$:

$$f(\beta_h) \approx f(\hat{\beta}_h) + \frac{1}{2}(\beta_h - \hat{\beta}_h)^T H(\hat{\beta}_h)(\beta_h - \hat{\beta}_h)$$

$H(\hat{\beta}_h) = \nabla^2 f(\hat{\beta}_h)$

- As $q(\beta_h) \propto \exp\{f(\beta_h)\}$,

$$q(\beta_h) \approx N(\hat{\beta}_h, -H(\hat{\beta}_h)^{-1})$$

- $\hat{\beta}_h$: general numerical optimization methods
- We use BFGS algorithm via `optim` in R

Nonconjugate variational message passing (NCVMP)

- Assume
  1. $q(\theta) = \prod_{i=1}^{m} q_i(\theta_i)$ for $\theta = \{\theta_1, \dots, \theta_m\}$ (VB)
  2. each $q_i(\theta_i)$ is a member of some exponential family:

$$q_i(\theta_i) = \exp\{\lambda_i^T t_i(\theta_i) - h_i(\lambda_i)\},$$

  λ_i: vector of natural parameters, $t_i(\cdot)$: sufficient statistics

- Fixed point update ($\nabla_{\lambda_i}\mathcal{L} = 0$ when $\mathcal{L}$ is maximized):

$$\lambda_i \leftarrow \mathrm{Cov}_{q_i}[t_i(\theta_i)]^{-1} \, \nabla_{\lambda_i} E_q\{\log p(y, \theta)\} \;\; \text{for} \;\; i = 1, \dots, m,$$

- Counter convergence issues using damping

Nonconjugate variational message passing

- Assume $q(\beta_h) = N(\mu_h, \Sigma_h)$
- NCVMP update (Wand 2014):

$$\Sigma_h \leftarrow -\frac{1}{2}\left[\text{vec}^{-1}\left(\frac{\partial E_q\{\log p(y,\theta)\}}{\partial \text{vec}(\Sigma_h)}\right)\right]^{-1}$$

$$\mu_h \leftarrow \mu_h + \Sigma_h \frac{\partial E_q\{\log p(y,\theta)\}}{\partial \mu_h}.$$

- Explicit updates reduces computational cost significantly
- Numerical optimization of full $K \times K$ covariance matrix Σ_h is expensive for large K.

Delta method for moments

- $E_q\{\log p(y,\theta)\}$ cannot be computed in closed form as

$$E_q\left[\log\left\{1_J^T\exp(x_{ht}^T\beta_h)\right\}\right] \qquad (1)$$

  is intractable.

- Quadrature is computationally intensive
- Braun & McAuliffe (2010): approximate (1) using Jensen's inequality or delta method for moments (restrict Σ_h to be diagonal)
- We approximate (1) using the delta method
- Consider full covariance matrix for Σ_h. Feasible as NCVMP is fast

Delta method for moments

- Let $g_t(\beta_h) = \log\left\{1_J^T \exp(x_{ht}^T \beta_h)\right\}$
- Approximate $g_t(\beta_h)$ with second order Taylor expansion at μ_h and take expectations

$$E_q\{g_t(\beta_h)\}$$
$$= \log\left\{1_J^T \exp(x_{ht}^T \mu_h)\right\} + \frac{1}{2}\mathrm{tr}\left\{x_{ht}^T \left(\mathrm{diag}(\rho_{ht}) - \rho_{ht}\rho_{ht}^T\right) x_{ht}\Sigma_h\right\}$$

- $\rho_{ht} = \frac{\exp(x_{ht}^T \mu_h)}{1_J^T \exp(x_{ht}^T \mu_h)}$

Delta method for moments

- Closed form updates for μ_h and Σ_h:

$$\Sigma_h \leftarrow \left\{ \sum_{t=1}^{T_h} x_{ht}^T \left(\operatorname{diag}(\rho_{ht}) - \rho_{ht}\rho_{ht}^T \right) x_{ht} + \omega\Upsilon^{-1} \right\}^{-1}$$

$$\begin{aligned} \mu_h \leftarrow \mu_h + \Sigma_h \Big[& -\omega\Upsilon^{-1}(\mu_h - \mu_\zeta) + \sum_{t=1}^{T_h} x_{ht}^T (y_{ht} - \rho_{ht}) \\ & + x_{ht}^T \left(\operatorname{diag}(\rho_{ht}) - \rho_{ht}\rho_{ht}^T \right) \left\{ x_{ht}\Sigma_h x_{ht}^T \rho_{ht} - \tfrac{1}{2}\operatorname{diag}(x_{ht}\Sigma_h x_{ht}^T) \right\} \Big] \end{aligned}$$

- Compute an approximation $\mathcal{L}^*$ of $\mathcal{L}$
- Good posterior estimation
- Convergence not guaranteed as $\mathcal{L}^*$ is not lower bound to $\log p(y)$

Stochastic linear regression

- Apply fixed-form variational Bayes to any posterior (closed form up to proportionality constant) without evaluating integrals analytically
- Assumptions: as in NCVMP
- NCVMP update:

$$\lambda_i = \text{Cov}_{q_i}[t_i(\theta_i)]^{-1}\ \text{Cov}_{q_i}\left[t_i(\theta_i), E_{-q_i}\{\log p(y,\theta)\}\right]$$

- Weighted Monte Carlo by generating random samples from $q_i(\theta_i)$
- $q_i(\theta_i) = N(\mu_i, \Sigma_i)$: $\Sigma_i = P_i^{-1}$ and $\mu_i = m_i + \Sigma_i g_i$
- $P_i = -E_{q_i}[\nabla^2_{\theta_i} E_{-q_i}\{\log p(y,\theta)\}]$, $g_i = E_{q_i}[\nabla_{\theta_i} E_{-q_i}\{\log p(y,\theta)\}]$, $m_i = E_{q_i}\{\theta_i\}$

Weighted Monte Carlo

Initialize μ_i, Σ_i, $g_i = 0$, $P_i = \Sigma_i^{-1}$, $m_i = \mu_i$, $\bar{m}_i = 0$, $\bar{P}_i = 0$ and $\bar{g}_i = 0$.

For $n = 1, \ldots, N$,

- Generate $\hat{\theta}_i$ from $N(\mu_i, \Sigma_i)$
- Compute gradient $\hat{g}_i$ and Hessian $\hat{H}_i$ of $E_{-q_i}\{\log p(y, \theta)\}$ at $\hat{\theta}_i$
- For $0 \leq w \leq 1$, $P_i \leftarrow (1-w)P_i - w\hat{H}_i$, $g_i \leftarrow (1-w)g_i + w\hat{g}_i$, $m_i \leftarrow (1-w)m_i + w\hat{\theta}_i$
- Compute new estimates: $\Sigma_i \leftarrow P_i^{-1}$ and $\mu_i \leftarrow m_i + \Sigma_i g_i$
- If $n > N/2$, $\bar{P}_i \leftarrow \bar{P}_i - \frac{2}{N}\hat{H}_i$, $\bar{g}_i \leftarrow \bar{g}_i + \frac{2}{N}\hat{g}_i$ and $\bar{m}_i \leftarrow \bar{m}_i + \frac{2}{N}\hat{\beta}_i$

Set $\Sigma_i = \bar{P}_i^{-1}$ and $\mu_h = \Sigma_i \bar{g}_i + \bar{m}_i$

Stochastic linear regression

- q_i updated continually
- Weights w: diminish effects from early iterations (q_i less accurate)
- Fixed weights, average iterates over second half of iterations to reduce variability
- Set N: balance between accuracy and efficiency
  - Large N: inefficient
  - Small N: $\{\mu_i, \Sigma_i\}$ not close to convergence, accuracy deteriorates
- Does not require use of delta method to approximate expectations
- Overcomes convergence issues in NCVMP (sufficiently small w ensures convergence)

Stochastic linear regression

- Combined approach:
  1. update $q(\beta_h)$ for $h = 1, \ldots, H$ using stochastic linear regression
  2. $q(\zeta)$, $q(\Omega)$ and $q(a)$: explicit variational parameter updates
- Straightforward extension to stochastic variational inference

Comparison of three approaches

Laplace variational inference	NCVMP	Stochastic linear regression
$q(\beta_h) \approx N(\mu_h, \Sigma_h)$	$q(\beta_h) \approx N(\mu_h, \Sigma_h)$	$q(\beta_h) \approx N(\mu_h, \Sigma_h)$
uses Laplace approximation within variational Bayes optimal density update	uses delta method to approximate intractable integrals	does not require evaluating integrals analytically
• optimizes only μ_h (location of Gaussian variational posterior) • Set Σ_h as negative inverse Hessian at this point • often underestimates standard deviation terms in Ω	• optimizes μ_h and Σ_h using closed form updates	• optimizes μ_h and Σ_h using weighted Monte Carlo

Algorithm 1

Set $b_k = \frac{\nu+K}{2}$ for $k = 1, \ldots, K$ and $\omega = H + \nu + K - 1$.

Initialize $\mu_\zeta = \mu_h = 0$, $\Sigma_\zeta = \Sigma_h = 0.01\, I_K$, $\Upsilon = (\omega - K + 1)\, I_K$, $c = b$.

Cycle:

- Update μ_h and Σ_h for $h = 1, \ldots, H$ using
  1. Laplace variational inference
  2. NCVMP
  3. Stochastic linear regression
- $\Sigma_\zeta \leftarrow \left(\Sigma_0^{-1} + H\omega\Upsilon^{-1}\right)^{-1}$, $\mu_\zeta \leftarrow \Sigma_\zeta \left(\Sigma_0^{-1}\mu_0 + \omega\Upsilon^{-1}\sum_{h=1}^{H}\mu_h\right)$
- $\Upsilon \leftarrow 2\nu\mathrm{diag}\left(\frac{b}{c}\right) + \sum_{h=1}^{H}\{(\mu_h - \mu_\zeta)(\mu_h - \mu_\zeta)^T + \Sigma_h\} + H\Sigma_\zeta$
- $c_k \leftarrow \nu\omega\Upsilon_{kk}^{-1} + \frac{1}{A_k^2}$ for $k = 1, \ldots, K$

until convergence

Stochastic variational inference

- Algorithm 1: Update $\{\mu_h, \Sigma_h\}$ for $h = 1, \ldots, H$, before re-estimating $\{\mu_\zeta, \Sigma_\zeta, \Upsilon, c\}$ at each iteration
- Procedure increasingly inefficient as H increases

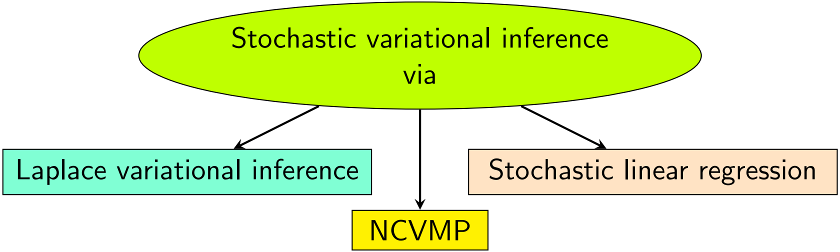
Stochastic variational inference (Hoffman et al. 2013)

At each iteration,

- draw a minibatch B of agents randomly from entire pool of agents
- Local variational parameters: Optimize μ_h and Σ_h for $h \in B$ (as a function of current global variational parameters)
- Global variational parameters: Stochastic natural gradient ascent (Robbins and Monroe 1951). Compute gradient estimates using optimized μ_h and Σ_h for $h \in B$

Stochastic variational inference

- Computation time reduced significantly when H is large.
- Large datasets in discrete choice modelling increasingly common
- Stochastic variational inference: important role in estimation


Stochastic variational inference
via
Laplace variational inference
NCVMP
Stochastic linear regression

Stochastic gradient ascent updates

- Gobal variational parameters
- At lth iteration,

$$\lambda_i^{(l+1)} = \lambda_i^{(l)} + \alpha_l \, \tilde{\nabla}_{\lambda_i}\mathcal{L}.$$

- $\tilde{\nabla}_{\lambda_i}\mathcal{L}$: natural gradient of $\mathcal{L}$ w.r.t λ_i
- Premultiply ordinary gradient $\nabla_{\lambda_i}\mathcal{L}$ with inverse of Fisher information matrix of $q_i(\theta_i)$ to obtain $\tilde{\nabla}_{\lambda_i}\mathcal{L}$
- $q_i(\theta_i)$: member of exponential family

$$\tilde{\nabla}_{\lambda_i}\mathcal{L} = \mathrm{Cov}_{q_i}[t_i(\theta_i)]^{-1} \, \nabla_{\lambda_i} E_q\{\log p(y, \theta)\} - \lambda_i$$

Stochastic gradient ascent updates

- λ_ζ, λ_Ω and λ_{β_h}: natural parameter vectors of $q(\zeta)$, $q(\Omega)$ and $q(\beta_h)$
- $\lambda_{\beta_h}^{\text{opt}}$: λ_{β_h} optimized as function of current global variational parameters

$$\tilde{\nabla}_{\lambda_\zeta}\mathcal{L} = \text{Cov}_{q(\zeta)}[t(\zeta)]^{-1}\nabla_{\lambda_\zeta}\Big[\sum_{h=1}^{H} E_q\{\log p(\beta_h|\zeta,\Omega)\}|_{\lambda_{\beta_h}=\lambda_{\beta_h}^{\text{opt}}} + E_q\{\log p(\zeta|\mu_0,\Sigma_0)\}\Big] - \lambda_\zeta$$

- B: minibatch of agents drawn randomly from entire pool of agents
- Unbiased estimate of $\tilde{\nabla}_{\lambda_\zeta}\mathcal{L}$: $\hat{\lambda}_\zeta - \lambda_\zeta$

$$\hat{\lambda}_\zeta = \text{Cov}_{q(\zeta)}[t(\zeta)]^{-1}\nabla_{\lambda_\zeta}\Big[\frac{H}{|B|}\sum_{h\in B} E_q\{\log p(\beta_h|\zeta,\Omega)\}|_{\lambda_{\beta_h}=\lambda_{\beta_h}^{\text{opt}}} + E_q\{\log p(\zeta|\mu_0,\Sigma_0)\}\Big]$$

Stochastic gradient ascent updates

- Unbiased estimate of $\tilde{\nabla}_{\lambda_\Omega}\mathcal{L}$: $\hat{\lambda}_\Omega - \lambda_\Omega$

$$\hat{\lambda}_\Omega = \mathrm{Cov}_{q(\Omega)}[t(\Omega)]^{-1}\nabla_{\lambda_\Omega}\Big[\frac{H}{|B|}\sum_{h\in B} E_q\{\log p(\beta_h|\zeta,\Omega)\}|_{\lambda_{\beta_h}=\lambda_{\beta_h}^{\mathrm{opt}}} + E_q\{\log p(\Omega|\nu,a)\}\Big]$$

- Stochastic gradient updates:

$\lambda_\zeta^{(l+1)} = (1-\alpha_l)\,\lambda_\zeta^{(l)} + \alpha_l\,\hat{\lambda}_\zeta$ and $\lambda_\Omega^{(l+1)} = (1-\alpha_l)\,\lambda_\Omega^{(l)} + \alpha_l\,\hat{\lambda}_\Omega.$

- Recover updates in Algorithm 1 when $|B| = H$ and $\alpha_l = 1$

Stochastic gradient ascent updates

- Iterates converge under certain regularity conditions (Spall 2003).
- Stepsizes: $\alpha_l \to 0$, $\sum_{l=0}^{\infty} \alpha_l = \infty$ and $\sum_{l=0}^{\infty} \alpha_l^2 < \infty$
- Common gain sequence: $\alpha_l = \frac{d}{(l+D)^{\gamma}}$, $0.5 < \gamma \leq 1$
- Stochastic approximation algorithms sensitive to rate of decrease of stepsizes. Tuning usually required
- Adaptive stepsize (Ranganath et al. 2013): minimize expected distance between stochastic and batch updates

Adaptive batch sizes

- Assume
  - large but finite number of agents H
  - possible to process dataset all at once (batch mode)
- Propose: Increase minibatch size adaptively as optimization proceeds until whole dataset is used
- Existing approach: keep minibatch size fixed, use decreasing stepsize to reduce noise

Adaptive batch sizes (Motivation)

- Beginning: estimates of global variational parameters are far from optimum. Only a small minibatch required to compute appropriate direction to move in.
- Estimates move closer towards optimum: more accurate definition of direction to move is required (use larger minibatches)
- Eventually: entire dataset is used. Convergence ensured, same level of accuracy attained as in batch mode
- Avoid having to specify a stopping criterion
  - Most criteria do not guarantee that terminal iterate is close to optimum and may be satisfied by chance
  - Termination often based on predetermined computational budget
- Avoid risk of iterates appearing to converge due to diminishing stepsizes

Previous methods

- Least mean squares (Orr 1996):
  - Derived formula for optimal minibatch size at each iteration
  - Results of theoretical interest but difficult to apply in practice
- L1-regularized problems (Boyles et al. 2011) and matrix factorization (Korattikara et al.1 2011):
  - Constructed frequentist hypothesis tests based on Central Limit Theorem to determine if parameter updates are in correct direction
  - Increase minibatch size when all parameters fail their tests
- Attempted hypothesis testing approach – Tests tend to fail too early

Proposed strategy for increasing minibatch sizes adaptively

- Perform stochastic variational inference with minibatches (size $|B|$)
- Update global variational parameters with constant stepsize:
  - No formal convergence
  - Popular practice as algorithm tends to be more robust
  - Iterates move monotonically towards optimum at first
  - Near the optimum, iterates bounce around instead of converge towards it as stepsizes remain large
- Oscillation: current minibatch size inadequate in providing direction to move
- More resolution required: increase minibatch size by a factor κ
- Repeat until whole dataset is used.

Detecting oscillation

Ratio of progress and path (Gaivoronski, 1988)

$$\phi^{(l)} = \frac{|\lambda_i^{(l-M)} - \lambda_i^{(l)}|}{\sum_{r=l-M}^{l-1} |\lambda_i^{(r)} - \lambda_i^{(r+1)}|}$$

for a univariate variable λ_i at iteration l

- $0 \leq \phi^{(l)} \leq 1$
- $\phi^{(l)} = 0$: no progress after M iterations
- $\phi^{(l)} = 1$: path from $\lambda_i^{(l-M)}$ to $\lambda_i^{(l)}$ is monotonic
- Small $\phi^{(l)}$: path is erratic, a lot of back and forth movement.
- Store $\lambda_i^{(l-M)}, \ldots, \lambda_i^{(l)}$ in memory for computing $\phi^{(l)}$
- Gaivoronski (1988) used this ratio to define an adaptive stepsize

Proposed strategy

- Monitor "ratio of progress and path" for elements in μ_ζ and diag(Υ)
- Set M=20. Compute ratios when $l > 5$ using available history
- If $5 < l < M$, $\phi_{1k}^{(l)} = \frac{|\Upsilon_{kk}^{(0)} - \Upsilon_{kk}^{(l)}|}{\sum_{r=0}^{l-1} |\Upsilon_{kk}^{(r)} - \Upsilon_{kk}^{(r+1)}|}$ and $\phi_{2k}^{(l)} = \frac{|{\mu_\zeta}_k^{(0)} - {\mu_\zeta}_k^{(l)}|}{\sum_{r=0}^{l-1} |{\mu_\zeta}_k^{(r)} - {\mu_\zeta}_k^{(r+1)}|}$
- If $l \geq M$, $\phi_{1k}^{(l)} = \frac{|\Upsilon_{kk}^{(l-M)} - \Upsilon_{kk}^{(l)}|}{\sum_{r=l-M}^{l-1} |\Upsilon_{kk}^{(r)} - \Upsilon_{kk}^{(r+1)}|}$ and $\phi_{2k}^{(l)} = \frac{|{\mu_\zeta}_k^{(l-M)} - {\mu_\zeta}_k^{(l)}|}{\sum_{r=l-M}^{l-1} |{\mu_\zeta}_k^{(r)} - {\mu_\zeta}_k^{(r+1)}|}$
- $\min \left\{ \phi_{1k}^{(l)}, \phi_{2k}^{(l)} \,|\, k = 1, \ldots, K \right\} < \Phi$: increase $|B|$ by factor κ
- Vary Φ with $|B|$. For small $|B|$, a smaller Φ is required as path of algorithm can be erratic even though progress is being made due to greater randomness between iterations

Algorithm 2 (1)

Set $b_k = \frac{\nu+K}{2}$ for $k = 1, \dots, K$ and $\omega = H + \nu + K - 1$.

Initialize $\mu_\zeta = \mu_h = 0$, $\Sigma_\zeta = \Sigma_h = 0.01\, I_K$, $\Upsilon = (\omega - K + 1)\, I_K$, $c = b$, $l = 0$ and $|B| = 25$.

While $|B| < H$,

- $l \leftarrow l + 1$
- Randomly select minibatch B of agents from entire pool of agents
- Optimize μ_h and Σ_h for $h \in B$ using
    - Laplace variational inference (as in Algorithm 1),
    - NCVMP (Cycle updates in Algorithm 1 until convergence), or
    - Stochastic linear regression (as in Algorithm 1)

Algorithm 2 (2)

- $\Sigma_\zeta \leftarrow \left(\Sigma_0^{-1} + H\omega\Upsilon^{-1}\right)^{-1}$,
  $\mu_\zeta \leftarrow \left(1-\alpha_{|B|}\right)\mu_\zeta + \alpha_{|B|}\Sigma_\zeta\left(\Sigma_0^{-1}\mu_0 + \omega\Upsilon^{-1}\frac{H}{|B|}\sum_{h\in B}\mu_h\right)$.
- $\Upsilon \leftarrow \left(1-\alpha_{|B|}\right)\Upsilon + \alpha_{|B|}\Big[2\nu\mathrm{diag}\left(\frac{b}{c}\right) + H\Sigma_\zeta$
  $+\frac{H}{|B|}\sum_{h\in B}\{(\mu_h-\mu_\zeta)(\mu_h-\mu_\zeta)^T + \Sigma_h\}\Big]$.
- $c_k \leftarrow \nu\omega\Upsilon_{kk}^{-1} + \frac{1}{A_k^2}$ for $k = 1,\dots,K$.
- If $l > 5$, compute $\phi_{1k}^{(l)}$ and $\phi_{2k}^{(l)}$ for $k = 1,\dots,K$.
  If $\min\left\{\phi_{1k}^{(l)}, \phi_{2k}^{(l)} \mid k = 1,\dots,K\right\} < \Phi_{|B|}$, $|B| \leftarrow \min\left\{\kappa|B|, H\right\}$,
  $l \leftarrow 0$

Algorithm 2 (3)

If $|B| = H$, cycle

- Update μ_h and Σ_h for $h = 1, \ldots, H$ using
    - Laplace approximation (as in Algorithm 1),
    - NCVMP (Cycle updates in Algorithm 1 until convergence in the first iteration and perform just once subsequently), or
    - stochastic linear regression (as in Algorithm 1)
- $\Sigma_\zeta \leftarrow \left(\Sigma_0^{-1} + H\omega\Upsilon^{-1}\right)^{-1}$, $\mu_\zeta \leftarrow \Sigma_\zeta \left(\Sigma_0^{-1}\mu_0 + \omega\Upsilon^{-1}\sum_{h=1}^{H}\mu_h\right)$.
- $\Upsilon \leftarrow 2\nu\mathrm{diag}\left(\frac{b}{c}\right) + \sum_{h=1}^{H}\{(\mu_h - \mu_\zeta)(\mu_h - \mu_\zeta)^T + \Sigma_h\} + H\Sigma_\zeta$.
- $c_k \leftarrow \nu\omega\Upsilon_{kk}^{-1} + \frac{1}{A_k^2}$ for $k = 1, \ldots, K$.

until convergence.

Algorithm 2

Note: Local variational parameters should be optimized as a function of current global variational parameters

- Laplace variational inference: optimizes only μ_h. Convergence ensured as entire dataset is used eventually
- NCVMP: updates for μ_h and Σ_h, are recursive, cycle until convergence
- Stochastic linear regression: Fix number of iterations at N and assume this is sufficient for $\{\mu_h, \Sigma_h\}$ to be close to convergence

Algorithm 2

- Use constant stepsizes within each minibatch size
- Allow stepsize $\alpha_{|B|}$ to increase with $|B|$
    - Beginning: smaller stepsizes required as we are less confident in the directions of gradient ascent computed using small minibatches of optimized local variational parameters
    - As minibatch size increases, confidence level increases
    - Stepsize is 1 when algorithm transits to batch mode ($|B| = H$)
- Start with $|B| = 25$, $\alpha_{|B|} = 0.4$ and $\Phi_{|B|} = 0.4$
- Increase $\alpha_{|B|}$ and $\Phi_{|B|}$ linearly with $|B|$ until they are 1 when $|B| = H$

Assessing proposed variational methods

- Improved random walk Metropolis algorithm for drawing β_h using fractional likelihood approach (Rossi et al. 2005)
  - Exhibits better mixing, dissipates initial conditions in shorter time than random walk Metropolis and independence Metropolis sampler
  - Implemented in `R` package `bayesm` via `rhierMnlRwMixture`
- Modify function to accommodate marginally noninformative priors for Ω
- Use this algorithm as basis for comparing MCMC with proposed variational methods

True predictive choice distribution

- True predictive choice distrbution of $y_{\text{new}(J\times 1)}$ given observed variables $x_{\text{new}(J\times K)}$:

$$p_{\text{true}}(y_{\text{new}}|x_{\text{new}},\zeta,\Omega) = \int p(y_{\text{new}}|x_{\text{new}},\beta)p(\beta|\zeta,\Omega)\ d\beta \qquad (2)$$

- Simulated data (ζ and Ω known): compute true predictive choice distribution using Monte Carlo integration by making 10^6 draws of β from $N(\beta|\zeta,\Omega)$ (variability not noticeable)

Estimated predictive choice distribution

- Point estimate of predictive choice distribution obtained by taking mean of (2) under posterior of ζ and Ω:

$$\hat{p}(y_{\text{new}}|x_{\text{new}}, y) = \int \left\{ \int p(y_{\text{new}}|x_{\text{new}}, \beta) p(\beta|\zeta, \Omega)\, d\beta \right\} \times p(\zeta, \Omega|y)\, d\zeta\, d\Omega$$

- Compute estimated predictive choice distribution using Monte Carlo integration for variational and MCMC methods
- Use 500 draws of $\{\zeta, \Omega\}$ from $q(\zeta)q(\Omega)$ for variational methods and 10000 draws from MCMC simulations
- More samples used in case of MCMC as draws are autocorrelated

Total variation (TV) metric

- Compute distance between two predictive choice distributions (Levin et al. 2009)
- Simulated data: TV distance between estimated and true predictive choice distributions at attribute matrix x_{new}:

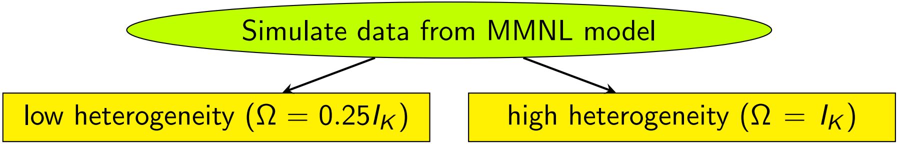
$$\begin{aligned}&\text{TV}[p_{\text{true}}(y_{\text{new}}|x_{\text{new}}), \hat{p}(y_{\text{new}}|x_{\text{new}})]\\&\qquad = \frac{1}{2}\sum_{j=1}^{J} |p_{true}(y^j_{\text{new}} = 1|x_{\text{new}}) - \hat{p}(y^j_{\text{new}} = 1|x_{\text{new}})|\end{aligned}$$

- Real data: true predictive choice distribution unknown
- Compute TV distances between predictive choice distributions estimated using MCMC and variational methods

Examples

- Stochastic linear regression (SLR):
  - Set $N = 40$ and $w = 0.25$
  - Algorithm 1: mean runtime and standard deviation over 5 runs
  - Algorithm 2: mean runtime and standard deviation over 10 runs
- MCMC
  - 4 independent chains, first half of each discarded as burn-in
  - Report average time taken to run a single chain and standard deviation over four chains
  - Gelman-Rubin diagnostics: 10000 draws that remained after thinning are good approximaton of posterior distribution

Simulated data



Simulate data from MMNL model

low heterogeneity ($\Omega = 0.25 I_K$)

high heterogeneity ($\Omega = I_K$)

- $H = 10000$ agents, $J = 12$ alternatives, $K = 10$ attributes
- $T_h = 25$ observed events for each agent h
- ζ: equally spaced values from -2 to 2
- Entries in x_{ht} generated independently from $N(0, 0.5^2)$
- MCMC: 10000 iterations in each chain, thinning factor: 2
- Algorithm 2: experimented with κ from 2 to 20. Larger κ led to greater reduction in computation time

Simulated data

Heterogeneity	Methods	Algorithm 1	Algorithm 2 ($\kappa = 20$)	Reduction
Low	Laplace	1008	470 (29)	53%
	NCVMP	432	311 (11)	28%
	SLR	1752 (5)	797 (51)	55%
	MCMC	23991 (152)	–	–
High	Laplace	1348	674 (51)	50%
	NCVMP	716	389 (42)	46%
	SLR	1752 (4)	1104 (110)	37%
	MCMC	24052 (250)	–	–

Table : CPU times (seconds) for MCMC and Algorithms 1 and 2. Last column indicates percentage reduction in CPU times from using Algorithm 2 instead of 1. Standard deviation over repeated runs in brackets.

- All variational methods are faster than MCMC by an order of magnitude

Simulated data

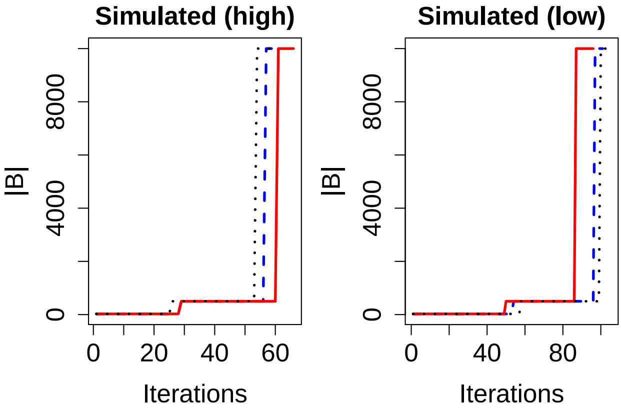


Figure : Plots show average number of iterations spent by Algorithm 2 at each minibatch size $|B|$. Blue dashed lines correspond to Laplace, red lines to NCVMP and black dotted lines to SLR.

Simulated data

Het.	Methods	Min.	1st Qu.	Median	Mean	3rd Qu.	Max
Low	Laplace	1.04%	2.04%	2.38%	2.38%	2.69%	4.02%
	NCVMP	0.14%	0.38%	0.50%	0.49%	0.60%	0.96%
	SLR	0.13%	0.34%	0.44%	0.45%	0.54%	0.92%
	MCMC	0.09%	0.37%	0.47%	0.47%	0.57%	0.89%
High	Laplace	0.63%	1.52%	1.76%	1.79%	2.04%	3.02%
	NCVMP	0.07%	0.31%	0.41%	0.44%	0.54%	1.00%
	SLR	0.04%	0.29%	0.41%	0.44%	0.57%	1.08%
	MCMC	0.04%	0.31%	0.42%	0.45%	0.56%	1.05%

Table : Summary of TV errors of MCMC and variational methods from true predictive choice distribution. TV errors computed at 500 attribute matrices x_{new} (entries generated randomly from $N(0, 0.5^2)$)

- Little difference in accuracy between NCVMP, SLR and MCMC, while Laplace approximation did much worse than the rest

Project on faculty appointments

- Subset of data from The Project on Faculty Appointments
- Study at Harvard Graduate School of Education to examine importance of different factors in job decisions (Trower 2002)
- Survey respondents ($H = 1274$ faculty and doctoral candidates) each presented with $T_h = 16$ pairs of job positions
- Select one of the two positions or neither, for each pair ($J = 3$)
- Positions varied along factors: balance of work, chance of tenure or contract renewal, geographic location, department rating, salary, institution rating, tenure or non-tenure track and length of contract for non-tenured track
- $K = 10$ covariates (effect coded indicator variables for factors described above, with two to four levels)

Project on faculty appointments

Methods	Algorithm 1	Algorithm 2 ($\kappa = 2$)	Reduction
Laplace	707	325 (57)	54%
NCVMP	113	51 (5)	55%
SLR	714 (12)	274 (39)	62%
MCMC	13719 (68)	–	–

Table : CPU times (seconds) for MCMC and Algorithms 1 and 2. Last column indicates percentage reduction in CPU times from using Algorithm 2 instead of 1. Standard deviation over repeated runs in brackets.

- MCMC: 50 000 iterations in each chain, thinning factor: 10 (Parameters for several variables took a long time to converge and there was high correlation between draws)
- Very good reductions of 54%–62% when using Algorithm 2 instead of 1
- All variational methods faster than MCMC by factor of 20–270.

Project on faculty appointments

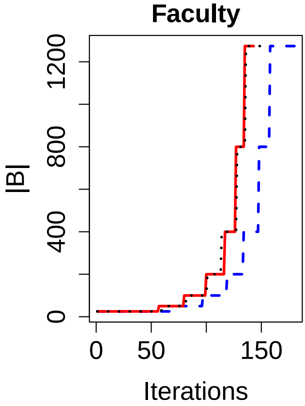


Figure : Plot shows average number of iterations spent by Algorithm 2 at each minibatch size $|B|$. Blue dashed lines correspond to Laplace, red lines to NCVMP and black dotted lines to SLR.

Project on faculty appointments

Methods	Min.	1st Qu.	Median	Mean	3rd Qu.	Max
Laplace vs. MCMC	0.45%	1.22%	1.80%	1.83%	2.21%	3.70%
NCVMP vs. MCMC	0.02%	0.13%	0.24%	0.24%	0.31%	0.57%
SLR vs. MCMC	0.02%	0.15%	0.22%	0.22%	0.28%	0.51%

Table : Summary of 1274 TV distances between predictive choice distribution estimated using MCMC and Algorithm 1. TV distances computed at 1274 attribute matrices, obtained by randomly selecting one covariate matrix x_{ht} from each respondent.

- SLR produced results closest to that of MCMC with NCVMP close behind
- Results Laplace much further away from MCMC than NCVMP and SLR

Electricity data

- $H = 361$ residential electricity customers were each presented with 12 choice experiments ($8 \leq T_h \leq 12$)
- Choose an electricity supplier out of $J = 4$ alternatives
- Attributes of suppliers: price, contract length in years and whether company was local or well-known
- K=6 covariates
- MCMC: 10000 iterations in each chain, thinning factor: 2

Electricity data

Methods	Algorithm 1	Algorithm 2 ($\kappa = 2$)	Reduction
Laplace	159	69 (4)	57 %
NCVMP	Diverge	Diverge	–
SLR	255 (6)	157 (8)	38 %
MCMC	756 (18)	–	–

Table : CPU times (seconds) for MCMC and Algorithms 1 and 2. Last column indicates percentage reduction in CPU times from using Algorithm 2 instead of 1. Standard deviation over repeated runs in brackets.

- NCVMP failed to converge because of delta method approximation
- SLR can overcome convergence issues in NCVMP + delta method
- As SLR is slower, one could run Algorithm 1 using NCVMP and then switch to SLR when lower bound fails to increase.
- Small dataset but speedups can still be obtained using Algorithm 2

Electricity data

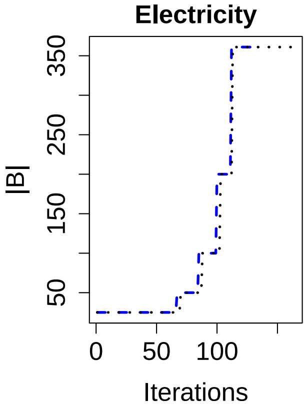


Figure : Plot shows average number of iterations spent by Algorithm 2 at each minibatch size $|B|$. Blue dashed lines correspond to Laplace and black dotted lines to SLR.

Electricity data

Methods	Min.	1st Qu.	Median	Mean	3rd Qu.	Max
Laplace vs. MCMC	0.88%	1.98%	2.31%	2.42%	3.18%	4.16%
SLR vs. MCMC	0.15%	0.36%	0.41%	0.43%	0.50%	0.73%

Table : Summary of 1444 TV distances between predictive choice probabilities computed using MCMC and Algorithm 1. TV distances computed at 1444 attribute matrices, obtained by randomly selecting four covariate matrices x_{ht} from each respondent.

- Very good agreement between SLR and MCMC
- Discrepancy between Laplace and MCMC much more pronounced

Conclusion

- Developed and investigated performances of three variational approaches for fitting MMNL models:
  1. Laplace approximation (Laplace)
  2. NCVMP + delta method
  3. stochastic linear regression (SLR)
- Accuracy: predictive inference from SLR closest to that of MCMC, with NCVMP close behind. Discrepancy between Laplace and MCMC much more pronounced
- Stability: SLR and Laplace are very stable. NCVMP failed to converge in one example due to the delta method
- Speed: NCVMP is fastest ($>$ 100 times speedup compared to MCMC)

Conclusion

- Stochastic variational inference accelerates convergence for large datasets
- Proposed a novel adaptive batch size strategy
  - Algorithm 2 is almost automatic
  - Increase κ proportionately with number of agents H
  - Significant speedups from Algorithm 2 for datasets as small as a few hundreds
- Variational methods: an important alternative and complement to MCMC methods for fitting MMNL models (high computational efficiency with competitive accuracy)[1]

[1] Tan, L. S. L. Stochastic variational inference for large-scale discrete choice models using adaptive batch sizes. arXiv:1405.5623