SUPPLEMENT TO: SHOPPER: A PROBABILISTIC MODEL OF CONSUMER CHOICE WITH SUBSTITUTES AND COMPLEMENTS

By Francisco J. R. Ruiz*,†, Susan Athey‡ and David M. Blei†

University of Cambridge, Columbia University,† and Stanford University‡*

1. Details on the Inference Algorithm. Here we provide the technical details of the variational inference procedure and a description of the full algorithm.

Recall the notation introduced in Section 5 of the main paper, where $\boldsymbol{\ell} = \{\rho, \alpha, \lambda, \theta, \gamma, \beta, \mu, \delta\}$ denotes the set of all latent variables in the model, $y = \{y_t\}$ is the collection of (unordered) baskets, and $x = x_{1:T}$, where $x_t = (u_t, w_t, r_t)$ are observed covariates of the shopping trips. We use mean-field variational inference, i.e., we approximate the posterior $p(\boldsymbol{\ell} \mid y, \mathbf{x})$ with a fully factorized variational distribution,

$$q(\boldsymbol{\ell}) = \prod_c q(\alpha_c) q(\rho_c) q(\lambda_c) q(\beta_c) q(\mu_c) \times \prod_u q(\theta_u) q(\gamma_u) \times \prod_w q(\delta_w). \tag{1}$$

We set each variational factor in the same family as the prior, i.e., Gaussian variational distributions with diagonal covariance matrices for $q(\rho_i)$, $q(\alpha_i)$, $q(\lambda_i)$, $q(\nu_u)$, $q(\mu_i)$, and $q(\delta_w)$, and independent gamma variational distributions for the price sensitivity terms $q(\gamma_u)$ and $q(\beta_i)$. We parameterize the Gaussian in terms of its mean and standard deviation, and we parameterize the gamma in terms of its shape and mean.

Let ν denote the vector containing all the variational parameters. We wish to find the variational parameters ν that maximize the evidence lower bound (ELBO),

$$\begin{aligned} \mathcal{L}(\nu) &= \mathbb{E}_{q(\boldsymbol{\ell}\,;\,\nu)} \left[\log p(y \mid \mathbf{x}, \boldsymbol{\ell}) + \log p(\boldsymbol{\ell}) - \log q(\boldsymbol{\ell}\,;\,\nu) \right] \\ &= \mathbb{E}_{q(\boldsymbol{\ell}\,;\,\nu)} \left[\sum_{t=1}^{T} \log p(y_t \mid x_t, \boldsymbol{\ell}) + \log p(\boldsymbol{\ell}) - \log q(\boldsymbol{\ell}\,;\,\nu) \right] \\ &\leq \log p(y \mid \mathbf{x}). \end{aligned} \tag{2}$$

In this supplement, we show how to apply stochastic optimization to maximize the bound on the log marginal likelihood. More in detail, we first describe how to tackle the intractable expectations using stochastic optimization and the reparameterization trick, and then we show how to leverage stochastic optimization to decrease the computational complexity.

1.1. *Intractable expectations: Stochastic optimization and the reparameterization trick.* We are interested in maximizing an objective function of the form

$$\widetilde{\mathcal{L}}(\nu) = \mathbb{E}_{q(\boldsymbol{\ell}\,;\,\nu)} \left[f(\boldsymbol{\ell}, \nu) \right] \tag{3}$$

with respect to the parameters ν. In the particular case that $f(\boldsymbol{\ell}, \nu) = \log p(y, \boldsymbol{\ell} \mid \mathbf{x}) - \log q(\boldsymbol{\ell} \,;\, \nu)$, we recover the ELBO in Eq. 2, but we prefer to keep the notation general because in Section 1.2 we will consider other functions $f(\boldsymbol{\ell}, \nu)$.

The main challenge is that the expectations in Eq. 3 are analytically intractable. Thus, the variational algorithm we develop aims at obtaining and following noisy estimates of the gradient $\nabla_\nu \widetilde{\mathcal{L}}$. We obtain these estimates via stochastic optimization; in particular, we apply the reparameterization trick to form Monte Carlo estimates of the gradient (Kingma and Welling, 2014; Titsias and Lázaro-Gredilla, 2014; Rezende, Mohamed and Wierstra, 2014).

In reparameterization, we first introduce a transformation of the latent variables $\boldsymbol{\ell} = \mathcal{T}(\boldsymbol{\epsilon} \,;\, \nu)$ and an auxiliary distribution $\pi(\boldsymbol{\epsilon} \,;\, \nu)$, such that we can obtain samples from the variational distribution $q(\boldsymbol{\ell} \,;\, \nu)$ following a two-step process:

$$\boldsymbol{\epsilon} \sim \pi(\boldsymbol{\epsilon} \,;\, \nu), \qquad \boldsymbol{\ell} = \mathcal{T}(\boldsymbol{\epsilon} \,;\, \nu). \tag{4}$$

The requirement for $\pi(\boldsymbol{\epsilon} \,;\, \nu)$ and $\mathcal{T}(\boldsymbol{\epsilon} \,;\, \nu)$ is that this procedure must provide a variable $\boldsymbol{\ell}$ that is distributed according to $\boldsymbol{\ell} \sim q(\boldsymbol{\ell} \,;\, \nu)$. Here we have considered the generalized reparameterization approach (Ruiz, Titsias and Blei, 2016; Naesseth et al., 2017), which allows the auxiliary distribution $\pi(\boldsymbol{\epsilon} \,;\, \nu)$ to depend on the variational parameters ν (this is necessary because the gamma random variables are not otherwise reparameterizable).

Once we have introduced the auxiliary variable $\boldsymbol{\epsilon}$, we can rewrite the gradient of the objective in Eq. 3 as an expectation with respect to the auxiliary distribution $\pi(\boldsymbol{\epsilon} \,;\, \nu)$,

$$\nabla_\nu \widetilde{\mathcal{L}} = \nabla_\nu \mathbb{E}_{q(\boldsymbol{\ell} \,;\, \nu)} \left[f(\boldsymbol{\ell}, \nu) \right] = \nabla_\nu \mathbb{E}_{\pi(\boldsymbol{\epsilon} \,;\, \nu)} \left[f\left(\mathcal{T}(\boldsymbol{\epsilon} \,;\, \nu), \nu \right) \right]. \tag{5}$$

We now push the gradient into the integral[1] and apply the chain rule for derivatives to express the gradient as an expectation,

$$\nabla_\nu \widetilde{\mathcal{L}} = \mathbb{E}_{\pi(\boldsymbol{\epsilon} \,;\, \nu)} \left[\nabla_{\boldsymbol{\ell}} f(\boldsymbol{\ell}, \nu) \big|_{\boldsymbol{\ell} = \mathcal{T}(\boldsymbol{\epsilon} \,;\, \nu)} \nabla_\nu \mathcal{T}(\boldsymbol{\epsilon} \,;\, \nu) + f\left(\mathcal{T}(\boldsymbol{\epsilon} \,;\, \nu), \nu \right) \nabla_\nu \log \pi(\boldsymbol{\epsilon} \,;\, \nu) \right]. \tag{6}$$

To obtain this expression, we have assumed that $\mathbb{E}_{\pi(\boldsymbol{\epsilon} \,;\, \nu)} \left[\nabla_\nu f(\boldsymbol{\ell}, \nu) \big|_{\boldsymbol{\ell} = \mathcal{T}(\boldsymbol{\epsilon} \,;\, \nu)} \right] = 0$ because the only dependence of $f(\boldsymbol{\ell}, \nu)$ on ν is through the term $\log q(\boldsymbol{\ell} \,;\, \nu)$, and the expectation of the score function is zero.

We can now obtain a Monte Carlo estimate of the expectation in Eq. 6 (and therefore of the gradient of interest) by drawing a sample from $\pi(\boldsymbol{\epsilon} \,;\, \nu)$ and evaluating the argument of the expectation. That is, we form the gradient estimator as

$$\widehat{\nabla}_\nu \widetilde{\mathcal{L}} = \nabla_{\boldsymbol{\ell}} f(\boldsymbol{\ell}, \nu) \big|_{\boldsymbol{\ell} = \mathcal{T}(\boldsymbol{\epsilon} \,;\, \nu)} \nabla_\nu \mathcal{T}(\boldsymbol{\epsilon} \,;\, \nu) + f\left(\mathcal{T}(\boldsymbol{\epsilon} \,;\, \nu), \nu \right) \nabla_\nu \log \pi(\boldsymbol{\epsilon} \,;\, \nu), \tag{7}$$

where $\boldsymbol{\epsilon} \sim \pi(\boldsymbol{\epsilon} \,;\, \nu)$. This assumes that we are able to evaluate $f(\boldsymbol{\ell}, \nu)$ and its gradient. (We show in Section 1.2 how to do that efficiently.)

[1] In the model that includes thinking-ahead, this step introduces a small bias due to the non-differentiability of the $\max(\cdot)$ operator; see Lee, Yu and Yang (2018).

Algorithm 1: Variational inference algorithm

input : Data y and x, model hyperparameters
output Variational parameters ν
:
Initialize ν randomly
Initialize iteration number $m \leftarrow 1$
repeat
  Sample $\boldsymbol{\epsilon} \sim \pi(\boldsymbol{\epsilon}\,;\,\nu)$
  Compute $\boldsymbol{\ell} = \mathcal{T}(\boldsymbol{\epsilon}\,;\,\nu)$
  Evaluate $f(\boldsymbol{\ell}, \nu)$ and $\nabla_{\boldsymbol{\ell}} f(\boldsymbol{\ell}, \nu)$ (see Algorithm 2)
  Obtain an estimate of the gradient, $\widehat{\nabla}_\nu \widetilde{\mathcal{L}}$ (Eq. 7)
  Set the step size $\eta^{(m)}$ (e.g., use the schedule proposed in ADVI)
  Take a gradient step, $\nu \leftarrow \nu + \eta^{(m)} \odot \widehat{\nabla}_\nu \widetilde{\mathcal{L}}$
  Increase the iteration number, $m \leftarrow m + 1$
until *convergence*
return ν

We use a transformation $\mathcal{T}(\cdot)$ and auxiliary distribution $\pi(\cdot)$ for each variational factor. For a Gaussian variational factor with mean μ and standard deviation σ, we use the standard reparameterization,

$$\mathcal{T}_{\text{Gauss}}(\varepsilon\,;\,\mu,\sigma) = \mu + \sigma\varepsilon, \qquad \pi_{\text{Gauss}}(\varepsilon\,;\,\mu,\sigma) = \mathcal{N}(0,1), \tag{8}$$

which makes the last term of Eq. 7 vanish because $\nabla_\nu \log \pi_{\text{Gauss}}(\varepsilon\,;\,\mu,\sigma) = 0$. For a gamma variational factor with shape α and mean μ, we use the transformation based on rejection sampling (Marsaglia and Tang, 2000),

$$\mathcal{T}_{\text{Gamma}}(\varepsilon\,;\,\alpha,\mu) = \frac{\mu}{\alpha}\left(\alpha - \frac{1}{3}\right)\left(1 + \frac{\varepsilon}{\sqrt{9\alpha - 3}}\right)^3, \tag{9}$$

and $\pi_{\text{Gamma}}(\varepsilon\,;\,\alpha,\mu)$ is defined though a rejection sampling procedure. See Naesseth et al. (2017) for further details about the reparameterization trick for gamma random variables.[2]

Algorithm 1 summarizes the resulting variational inference procedure. At each iteration, we obtain a sample from $\boldsymbol{\ell}$ via the auxiliary distribution $\pi(\boldsymbol{\epsilon}\,;\,\nu)$ and the transformation $\mathcal{T}(\boldsymbol{\epsilon}\,;\,\nu)$; we evaluate the function $f(\boldsymbol{\ell}, \nu)$ and its gradient with respect to the latent variables $\boldsymbol{\ell}$; we obtain the gradient estimate in Eq. 7; and we take a gradient step for the variational parameters ν. In the stochastic optimization procedure, we adaptively set the step size as proposed in the automatic differentiation variational inference (ADVI) algorithm (Kucukelbir et al., 2017).

1.2. *Computational complexity: Stochastic optimization and lower bounds.* The algorithm in Section 1.1 requires to evaluate the model log joint (as well as its gradient). There are three issues that make it expensive to evaluate the log joint. First, evaluating the log likelihood is expensive because it involves a summation

[2] In particular, we also apply the "shape augmentation trick," which allows us to reparameterize a gamma random variable with shape α in terms of another gamma random variable with shape $\alpha + P$, where P is a positive integer. We use $P = 10$. See Naesseth et al. (2017) for additional details.

over shopping trips. This represents a problem when the dataset is large. Second, evaluating the softmax involves computing its normalization constant, which contains a summation over all items. This becomes an issue when there are thousands of items and we need to evaluate many softmax probabilities. Third, computing the probability over unordered baskets y_t is also expensive, as it involves a summation over all possible permutations.

We address these issues by combining two techniques: data subsampling and lower bounds on the ELBO. We first describe data subsampling for evaluating the log likelihood. Note that the log likelihood involves a summation over many terms,

$$\log p(y \mid \mathbf{x}, \boldsymbol{\ell}) = \sum_{t=1}^{T} \log p(y_t \mid x_t, \boldsymbol{\ell}). \tag{10}$$

We can obtain an unbiased estimate of the log likelihood (and its gradient) by sampling a random subset of shopping trips. Let $\mathcal{B}_T$ be the (randomly chosen) set of trips. The estimator

$$\frac{T}{|\mathcal{B}_T|} \sum_{t \in \mathcal{B}_T} \log p(y_t \mid x_t, \boldsymbol{\ell})$$

is unbiased because its expected value is the log likelihood $\log p(y \mid \mathbf{x}, \boldsymbol{\ell})$ (Hoffman et al., 2013). Thus, we subsample data terms to obtain unbiased estimates of the log likelihood and its gradient, resulting in a computationally more efficient algorithm.

Second, we describe how to form variational bounds to address the issue of the expensive normalization constant of the softmax. Each softmax log probability is given by

$$\log p(y_{ti} = c \mid \mathbf{y}_{t,i-1}) = \Psi(c, \mathbf{y}_{t,i-1}) - \log \left(\sum_{c' \notin \mathbf{y}_{t,i-1}} \exp\{\Psi(c', \mathbf{y}_{t,i-1})\} \right). \tag{11}$$

The summation over c' is expensive, and we cannot easily form an unbiased estimator because of the non-linearity introduced by the logarithm. Hence, we apply the one-vs-each bound (Titsias, 2016), which allows us to write

$$\log p(y_{ti} = c \mid \mathbf{y}_{t,i-1}) \geq \sum_{c' \notin [\mathbf{y}_{t,i-1}, c]} \log \sigma\big(\Psi(c, \mathbf{y}_{t,i-1}) - \Psi(c', \mathbf{y}_{t,i-1})\big), \tag{12}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. We can form unbiased estimates of the summation via subsampling. More precisely, we randomly sample a set $\mathcal{B}_C^{(t,i)}$ of items (each of them distinct from c and from the other items in the basket). Then, we form the following unbiased estimator:

$$\frac{C - i}{|\mathcal{B}_C^{(t,i)}|} \sum_{c' \in \mathcal{B}_C^{(t,i)}} \log \sigma\big(\Psi(c, \mathbf{y}_{t,i-1}) - \Psi(c', \mathbf{y}_{t,i-1})\big).$$

Here, C stands for the total number of items.

Finally, we show how to deal with the issue of unordered baskets. Recall that each log likelihood term involves a summation over all possible permutations of the baskets (holding the checkout item in the last position),

$$\log p(y_t \mid x_t, \boldsymbol{\ell}) = \log \left(\sum_{\pi_t} p(\mathbf{y}_{t,\pi_t} \mid x_t, \boldsymbol{\ell}) \right). \tag{13}$$

Following a similar procedure as Doshi-Velez et al. (2009), we introduce an auxiliary distribution $q(\pi_t)$ to rewrite the expression above as an expectation with respect to $q(\pi_t)$, and then we apply Jensen's inequality:

$$\begin{aligned} \log p(y_t \mid x_t, \boldsymbol{\ell}) &= \log \left(\mathbb{E}_{q(\pi_t)} \left[\frac{p(\mathbf{y}_{t,\pi_t} \mid x_t, \boldsymbol{\ell})}{q(\pi_t)} \right] \right) \\ &\geq \mathbb{E}_{q(\pi_t)} \left[\log p(\mathbf{y}_{t,\pi_t} \mid x_t, \boldsymbol{\ell}) - \log q(\pi_t) \right] \\ &= \sum_{\pi_t} q(\pi_t) \left(\log p(\mathbf{y}_{t,\pi_t} \mid x_t, \boldsymbol{\ell}) - \log q(\pi_t) \right). \end{aligned} \tag{14}$$

Since the bound involves a direct summation over permutations, we can subsample terms to alleviate the computational complexity. For simplicity, we set $q(\pi_t)$ to be a uniform distribution over all possible permutations, and thus we do not introduce auxiliary variational parameters that would be too expensive to obtain otherwise. In particular, we form an unbiased estimate of the bound by sampling one random permutation π_t and evaluating the term[3]

$$\log p(\mathbf{y}_{t,\pi_t} \mid x_t, \boldsymbol{\ell}).$$

To sum up, we have derived a bound of the ELBO,

$$\widetilde{\mathcal{L}}(\nu) \leq \mathcal{L}(\nu) \leq \log p(y \mid x), \tag{15}$$

which can still be written as an expectation with respect to the variational distribution $q(\boldsymbol{\ell} \mid \nu)$. More importantly, we can efficiently evaluate the argument $f(\boldsymbol{\ell}, \nu)$ of such expectation and its gradient with respect to the latent variables $\boldsymbol{\ell}$.

Putting all together, the function $f(\boldsymbol{\ell}, \nu)$ that we use is given by

$$\begin{aligned} f(\boldsymbol{\ell}, \nu) = &\sum_{t=1}^{T} \sum_{\pi_t} q(\pi_t) \sum_{i=1}^{n_t} \sum_{c' \notin [\mathbf{y}_{t,i-1}, c]} \log \sigma \big(\Psi(c, \mathbf{y}_{t,i-1}) - \Psi(c', \mathbf{y}_{t,i-1}) \big) \\ &+ \log p(\boldsymbol{\ell}) - \log q(\boldsymbol{\ell} \,;\, \nu). \end{aligned} \tag{16}$$

We obtain an unbiased estimate via subsampling shopping trips, one permutation π_t for each one, and items c'. We use $|\mathcal{B}_T| = 100$ trips and $|\mathcal{B}_C^{(t,i)}| = 50$ items in our experiments.

Algorithm 2 outlines the procedure to obtain an unbiased estimate of $f(\boldsymbol{\ell}, \nu)$ for a given sample of the latent parameters, $\boldsymbol{\ell} \sim q(\boldsymbol{\ell} \,;\, \nu)$. Differentiation though Algorithm 2 gives the gradient $\nabla_{\boldsymbol{\ell}} f(\boldsymbol{\ell}, \nu)$, which is also required in the inference procedure (Algorithm 1).

[3] We ignore the term $\log q(\pi_t)$ because it is a constant.

Algorithm 2: Estimate of $f(\boldsymbol{\ell}, \nu)$ (Eq. 16)

input : Data y and x, a sample of the latent parameters $\boldsymbol{\ell}$, variational distribution $q(\boldsymbol{\ell} \,;\, \nu)$

output : An unbiased estimate of $f(\boldsymbol{\ell}, \nu)$ and its gradient (for the gradient, differentiate through the algorithm)

Initialize $\widehat{f} \leftarrow \log p(\boldsymbol{\ell}) - \log q(\boldsymbol{\ell} \,;\, \nu)$

Sample a set of baskets $\mathcal{B}_T \subseteq \{1, \ldots, T\}$

for $t \in \mathcal{B}_T$ **do**

  Sample a permutation π_t of the items in basket t

  Set $\mathbf{y}_t$ to the vector containing the items in basket t ordered according to π_t

  for $i = 1, \ldots, n_t$ **do**

    Set c to the ith item in $\mathbf{y}_t$

    Sample a set of items $\mathcal{B}_C^{(t,i)} \subseteq \{1, \ldots, C\} \setminus \{\mathbf{y}_{t,i-1}, c\}$

    for $c' \in \mathcal{B}_C^{(t,i)}$ **do**

      Update $\widehat{f} \leftarrow \widehat{f} + \frac{T}{|\mathcal{B}_T|} \times \frac{C-i}{|\mathcal{B}_C^{(t,i)}|} \times \log \sigma\big(\Psi(c, \mathbf{y}_{t,i-1}) - \Psi(c', \mathbf{y}_{t,i-1})\big)$

    end

  end

end

return $\widehat{f}$ and $\nabla_{\boldsymbol{\ell}} \widehat{f}$

References.

Doshi-Velez, F., Miller, K. T., Van Gael, J. and Teh, Y. W. (2009). Variational inference for the Indian buffet process. In *Proceedings of the International Conference on Artificial Intelligence and Statistics* **12**.

Hoffman, M. D., Blei, D. M., Wang, C. and Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research* **14** 1303–1347.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. In *International Conference on Learning Representations*.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A. and Blei, D. M. (2017). Automatic differentiation variational inference. *Journal of Machine Learning Research* **18** 1–45.

Lee, W., Yu, H. and Yang, H. (2018). Reparameterization gradient for non-differentiable models. In *Advances in Neural Information Processing Systems*.

Marsaglia, G. and Tang, W. W. (2000). A simple method for generating gamma variables. *ACM Transactions on Mathematical Software* **26** 363–372.

Naesseth, C., Ruiz, F. J. R., Linderman, S. and Blei, D. M. (2017). Reparameterization gradients through acceptance-rejection methods. In *Artificial Intelligence and Statistics*.

Rezende, D. J., Mohamed, S. and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*.

Ruiz, F. J. R., Titsias, M. K. and Blei, D. M. (2016). The generalized reparameterization gradient. In *Advances in Neural Information Processing Systems*.

Titsias, M. K. (2016). One-vs-each approximation to softmax for scalable estimation of probabilities. In *Advances in Neural Information Processing Systems*.

Titsias, M. K. and Lázaro-Gredilla, M. (2014). Doubly stochastic variational Bayes for non-conjugate inference. In *International Conference on Machine Learning*.