Bayesian Models for Product Size Recommendations

Vivek Sembium
Amazon
India
viveksem@amazon.com

Rajeev Rastogi
Amazon
India
rastogi@amazon.com

Lavanya Tekumalla
Amazon
India
lavanya@amazon.com

Atul Saroop
Amazon
India
asaroop@amazon.com

ABSTRACT

Lack of calibrated product sizing in popular categories such as apparel and shoes leads to customers purchasing incorrect sizes, which in turn results in high return rates due to fit issues. We address the problem of product size recommendations based on customer purchase and return data. We propose a novel approach based on Bayesian logit and probit regression models with ordinal categories {`Small, Fit, Large`} to model size fits as a function of the difference between latent sizes of customers and products. We propose posterior computation based on mean-field variational inference, leveraging the Polya-Gamma augmentation for the logit prior, that results in simple updates, enabling our technique to efficiently handle large datasets. Our Bayesian approach effectively deals with issues arising from noise and sparsity in the data providing robust recommendations. Offline experiments with real-life shoe datasets show that our model outperforms the state-of-the-art in 5 of 6 datasets. and leads to an improvement of 17-26% in AUC over baselines when predicting size fit outcomes.

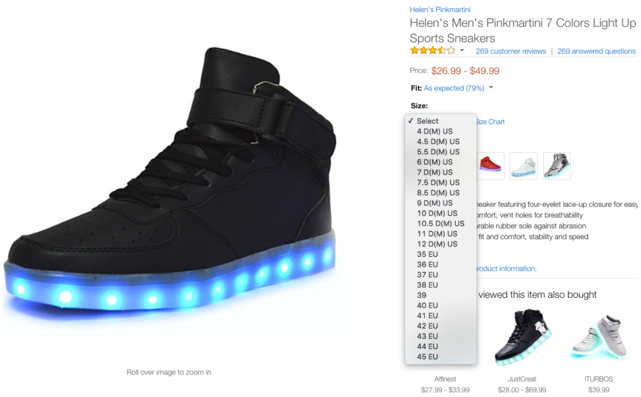
KEYWORDS

Bayesian, Personalization, Recommendation, Polya-Gamma, Probit, Variational Inference, Gibbs Sampling

ACM Reference Format:
Vivek Sembium, Rajeev Rastogi, Lavanya Tekumalla, and Atul Saroop. 2018. Bayesian Models for Product Size Recommendations. In *WWW 2018: The 2018 Web Conference, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3178876.3186149

1 INTRODUCTION

Recommending product sizes to customers is an important problem in the e-commerce domain. Though e-commerce is becoming increasingly popular, products such as apparel and shoes remain challenging to buy online and record high return rates. A key customer pain point that leads to excessive product returns is the *size fit problem.* Choosing the right size online is hard due to multiple reasons. Apart from the lack of touch and feel experience, variability in product sizing across brands and product types is a major challenge when shopping online. Essentially, brands and product types follow different sizing conventions, leading to different mappings from catalog sizes to physical sizes. For instance, the catalog size to physical size mappings for Reebok are: 6 = 15cm, 7 = 17cm, 8 = 21cm, while for Nike they are: 6 = 16cm, 7 = 18cm, 8 = 22cm. Further, there is variability in sizing even within the same brand. For instance, the running shoes of Nike have a different sizing scheme compared to walking shoes. Similarly, general running shoes have different sizing standards compared to those targeted towards marathon runners. Finally, incomplete and incorrect size charts provided by several sellers exacerbate this problem, making it difficult for customers to pick the correct size.



Helen's Pinkmartini
Helen's Men's Pinkmartini 7 Colors Light Up Sports Sneakers
269 customer reviews | 269 answered questions
Price: $26.99 - $49.99
Fit: As expected (79%)
Size: Select

neaker featuring four-eyelet lace-up closure for easy
omfort, vent holes for breathability
urable rubber sole against abrasion
fit and comfort, stability and speed

roduct information.

viewed this item also bought

Affinest $27.99 - $33.99
JustCreat $28.00 - $69.99
iTURBOS $39.99

Figure 1: Customer selecting a size variant on a shoe web page.

In the context of recommending product sizes, a customer's interaction with the e-commerce website takes place in the following manner: The customer visits the web page of a product and is provided with the option of choosing a particular size variant from several different sizes as shown in Figure 1. The *size recommendation problem* involves recommending the product size that would best fit the customer, thus improving customer experience. While there are several techniques [2, 9, 15, 18, 29] that address the general product recommendation problem, there is very little prior work related to generating size recommendations.

WWW 2018, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5639-8/18/04.
https://doi.org/10.1145/3178876.3186149

There are two inhibiting factors in generating accurate size recommendations. One, we do not have knowledge of the *customer's true size*. Two, accurate measurements of *product true sizes* are not available for all size variants. Nevertheless, valuable information about the true size of a customer lies in the *purchase history* of all the products previously purchased by the customer. Similarly, information about the true size of a product is available in the set of customers who have purchased it in the past. Another key piece of valuable information is the *return history*, which indicates if a product in the purchase history fit the customer or was returned due to incorrect choice of size.

However, recommending product sizes from purchase and return history is hard due to multiple reasons. Most e-commerce customers are occasional purchasers. Even the more frequent purchasers are unlikely to buy a large number of items from a specific category such as shoes. Hence, a majority of the customers have very few associated transactions. Similarly popular products tend to record huge sales, while a majority of products have very few or no associated transactions. As a consequence, customer transaction data is *highly sparse*. Further, return data is *highly noisy* as customers often do not indicate the right fitment information when returning an item. In fact, we have found a significant disagreement between the customer's input on size fit while returning an item and the customer's response to an independent fit related survey. Data sparsity and noise constitute a major impediment to making accurate size recommendations to the customer.

The inherent data sparsity necessitates a technique that enables learning the customer and product true size in a limited data setting, while the uncertainty arising from noisy purchase and return data necessitates computation of robust confidence estimates over the learned customer and product sizes to make confident recommendations. Motivated by this, we pursue a Bayesian treatment of the problem.

We propose a latent factor model that leverages a wide range of signals such as historical product purchases, returns by customers and product sizing information in the catalog to infer true (latent) sizes for customers and products. We propose Bayesian logit and probit regression models with ordinal categories to model the size fit data, which allows us to capture different *fitSuitabilityCodes* such as `Small, Fit` or `Large` provided by the customer. We propose efficient mean-field variational inference methods that leverage auxiliary variable techniques such as the Polya-Gamma augmentation to approximate the posterior distribution over customer and product true sizes, and use the posterior distribution to estimate the probability that a product with a particular size variant will fit the customer. Such a Bayesian treatment enables us to handle data sparsity issues by placing priors on customer and product true sizes based on catalog data, at the same time enabling computation of confidence intervals over the posterior distribution to propose confident recommendations despite noisy training data. In experiments with real-life shoe datasets, we show that our Bayesian models leveraging return information provided by customers have 17-26% higher AUC compared to baselines when predicting *fitSuitabilityCode* outcomes of purchase transactions.

The rest of the paper is structured as follows. In Section 2, we review related work, while in Section 3, we formally define the size recommendation problem. In Section 4, we propose our Bayesian model for size recommendations that learns the latent size for customers and products, with efficient posterior computation based on mean-field variational inference. In Section 5, we evaluate our models on real-world Amazon shoe datasets and synthetically generated data, and show a significant improvement over baselines. In Section 6, we discuss various extensions to our models and provide concluding remarks in Section 7.

2 RELATED WORK

While several techniques have been proposed to solve the general recommendation problem [3], there is very little work on size recommendation. Predominant amongst the techniques to solve the general recommendation problem, and most related to our work, are latent factor models based on Matrix Factorization [15, 18, 19, 29]. There have been extensions that model general non-Gaussian distributions [5], capture varying user preferences [7, 26, 27], are based on tensor factorization [16, 22, 27] and bilinear models [2, 9]. In [11, 14, 28], the authors take Bayesian approach to solve the recommendations problem and [20, 31] leverage user reviews, product metadata and other information. There are also approaches that leverage other contextual information, including physical context such as age, weather and device information [1, 10, 23, 24]. However, the above techniques for the general recommendation problem are not suitable for size recommendations, as they do not leverage additional ordinal information such as (1) catalog size of products and (2) customer provided *fitSuitability* outcome for the purchased product.

In this paper, we build Bayesian models that exploit size semantics to predict the ordinal outcome (`Small`, `Fit`, `Large`) based on the difference between latent sizes of the customer and the product involved in a transaction. We note that there has been prior work that explores a Bayesian treatment of the ordinal regression problem [4, 8, 12, 21, 25]. Our model differs from the usual setting of ordinal regression, since the predictor variable (the size difference between the customer and product) is itself latent. Our proposed inference strategies leverage state-of-the-art auxiliary variable techniques such as the Polya-Gamma augmentation [25] in a variational inference setting leading to a simple and efficient algorithm for computing the posterior.

The only prior work to the best of our knowledge in the academic literature that also addresses the specific problem of size recommendation is [30]. In [30], the authors propose a non-Bayesian approach that formulates the learning problem for customer and product true sizes as one of minimizing a carefully constructed loss function. The loss function minimization is carried out by an iterative algorithm that propagates updates between customer and product sizes through the purchase graph based on customer return instances. However the deterministic approach of [30], learns point estimates of product and customer true sizes that may have high errors due to the noisy return codes supplied by customers, exacerbated by extreme data sparsity issues. This in turn can lead to incorrect recommendations. Our Bayesian treatment of this problem enables us to place priors on customer and product sizes and get a posterior on these latent sizes over which we average to obtain more robust fit probability estimates (along with confidence intervals) to make accurate recommendations. We experimentally compare our technique with that of [30] in section 5 and outperform their technique in 5 of 6 datasets.

3 SIZE RECOMMENDATION PROBLEM

We now formally describe the size recommendation problem. A product is offered in multiple sizes, with the original product referred to as the *parent product*, while its various size variants are referred as *child products*. Let C denote the set of all customers and P the set of all child products in the catalog. The data comprises past customer transactions in the form of triplets (customer, child product, *fitSuitabilityCode*). Formally, let $D = (i, j, y_{ij})$ denote the transaction triplet data, where child product j was purchased by customer i and y_{ij} is `Fit` if the child product was not returned by the customer, or the *fitSuitabilityCode* `Small` or `Large` provided by the customer if it was returned by the customer.

Our objective is to recommend the child product with the best size fit for a specific (customer, parent product) pair. We do this by inferring posterior distribution over the true (latent) sizes of customers and child products using past transactions in D. Let s_i be the latent true size of a customer i and t_j be the latent true size of child product j. Intuitively, (1) if $(i, j, \texttt{Fit}) \in D$, then s_i should be as close to t_j as possible or $|s_i - t_j| \to 0$; (2) if $(i, j, \texttt{Small}) \in D$, then $(s_i - t_j)$ should be greater than a threshold $b_1 > 0$; and (3) if $(i, j, \texttt{Large}) \in D$, then $(s_i - t_j)$ should less than a threshold $b_2 < 0$. Once we have the true size estimates, we recommend the child product j with the highest size fit probability to customer i.

We adopt a Bayesian approach that leads to the following formulation for our size recommendation problem:

Problem Statement: Given past customer transactions D, catalog sizes $\{c_j\}$ for child products, and priors on true sizes, compute the posterior distribution on true sizes $\{s_i\}$ for customers and $\{t_j\}$ for child products. For a customer i and child product j, use the computed true size posteriors to estimate the size fit probability $P(y_{ij} = \texttt{Fit}|D)$. □

To simplify the presentation in the remainder of the paper, we will refer to child products as simply products.

4 BAYESIAN SIZE RECOMMENDATION MODEL

In this section, we present our probabilistic model, and schemes for learning the posterior distributions of customer true sizes $\{s_i\}$ and product true sizes $\{t_j\}$ and leveraging them to estimate the fit size probability $P(y_{ij} = \texttt{Fit}|D)$. We consider customer and product true sizes to be single dimensional and each account to belong to a single person. Generalizations of our techniques to handle multi-dimensional size vectors and multiple personas per customer account can be found in Section 6.

4.1 Data Likelihood

Let $f_\alpha(s_i, t_j) = \alpha \cdot (s_i - t_j)$ be a parametric function equal to the difference in true sizes between customer (s_i) and product (t_j) with a scale parameter α. Based on the earlier discussion in Section 3, we want the data likelihood $P(y_{ij}|s_i, t_j)$ for *fitSuitabilityCode* y_{ij} to satisfy: (1) if $f_\alpha(s_i, t_j) \geq b_1$ then $P(y_{ij} = \texttt{Small}|s_i, t_j)$ is large; (2) if $f_\alpha(s_i, t_j) \leq b_2$ then $P(y_{ij} = \texttt{Large}|s_i, t_j)$ is large; and (3) if $b_2 < f_\alpha(s_i, t_j) < b_1$ then $P(y_{ij} = \texttt{Fit}|s_i, t_j)$ is large. Below, we define a data likelihood function with the desired properties.

Let latent size vector $\beta = (s_1, ..., s_c, t_1, .., t_p, b_1, b_2)^T$, and for a transaction $(i, j, .) \in D$, define data vectors $x_{ijs} = (0, .., 0, \alpha, 0, .., 0, -\alpha, 0, .., 0, -1, 0)^T$ and $x_{ijf} = (0, .., 0, \alpha, 0, .., 0, -\alpha, 0, .., 0, 0, -1)^T$, where α and $-\alpha$ are present in the positions of s_i and t_j, respectively. Notice that learning the posterior of $\{s_i\}$ and $\{t_j\}$ is equivalent to learning the posterior of β. Also, $\beta^T \cdot x_{ijs} = f_\alpha(s_i, t_j) - b_1$ and $\beta^T \cdot x_{ijf} = f_\alpha(s_i, t_j) - b_2$.

To define the likelihood function for the three classes corresponding to the *fitSuitabilityCodes*, we introduce two binary random variables $y_{ijs} \in \{0, 1\}$ and $y_{ijf} \in \{0, 1\}$ with the following probability distributions:

$$P(y_{ijs}|s_i, t_j, \alpha, b_1) = \sigma(y_{ijs}, \beta^T \cdot x_{ijs}) \tag{1}$$

$$P(y_{ijf}|s_i, t_j, \alpha, b_2) = \sigma(y_{ijf}, \beta^T \cdot x_{ijf})$$

$$\text{where, } \sigma(y, \theta) = \frac{e^{y\theta}}{1 + e^{\theta}} \tag{2}$$

We note that an alternate choice for the link function σ is the Probit, where $P(y_{ijs} = 1|s_i, t_j, \alpha, b_1) = \Phi(\beta^T \cdot x_{ijs})$ and $P(y_{ijf} = 1|s_i, t_j, \alpha, b_1) = \Phi(\beta^T \cdot x_{ijf})$, where Φ is the CDF of the normal distribution. The probit approach is explored in detail in the Appendix.

We now define the likelihoods of `Small`, `Fit` and `Large` using these random variables as follows:

$$P(y_{ij} = \texttt{Small}|s_i, t_j, \alpha, b_1) = P(y_{ijs} = 1|\beta^T \cdot x_{ijs}) \tag{3}$$

$$\begin{aligned} P(y_{ij} = \texttt{Fit}|s_i, t_j, \alpha, b_1, b_2) \\ = P(y_{ijs} = 0|\beta^T \cdot x_{ijs}) \cdot P(y_{ijf} = 1|\beta^T \cdot x_{ijf}) \end{aligned} \tag{4}$$

$$\begin{aligned} P(y_{ij} = \texttt{Large}|s_i, t_j, \alpha, b_1, b_2) \\ = P(y_{ijs} = 0|\beta^T \cdot x_{ijs}) \cdot P(y_{ijf} = 0|\beta^T \cdot x_{ijf}) \end{aligned} \tag{5}$$

Notice that,

1. If $s_i >> t_j$, then $f_\alpha(s_i, t_j) - b_1 >> 0$. Then, $P(y_{ij} = \texttt{Small}|.) \to 1$.
2. If $s_i << t_j$, then $f_\alpha(s_i, t_j) - b_1 << 0$ and $f_\alpha(s_i, t_j) - b_2 << 0$. Then, $P(y_{ij} = \texttt{Large}|.) \to 1$.
3. If $b_2 < f_\alpha(s_i, t_j) < b_1$. Then, $P(y_{ij} = \texttt{Fit}|.) \to 1$.
4. $P(y_{ij} = \texttt{Small}|.) + P(y_{ij} = \texttt{Fit}|.) + P(y_{ij} = \texttt{Large}|.) = 1$.

4.2 Generative Model

We define our generative model as follows:

1. For each customer i, draw true size $s_i \sim \mathcal{N}(\mu_i, \sigma_s^2)$.
2. For each product j, draw true size $t_j \sim \mathcal{N}(c_j, \sigma_t^2)$.
3. Draw model parameter $b_1 \sim \mathcal{N}(\mu_{b_1}, \sigma_{b_1}^2)$ and model parameter $b_2 \sim \mathcal{N}(\mu_{b_2}, \sigma_{b_2}^2)$.
4. For each transaction $(i, j, y_{ij}) \in D$:
   - Select $y_{ij} = \texttt{Small}$ with probability $P(y_{ij} = \texttt{Small}|s_i, t_j, \alpha, b_1)$ defined in Equation (3).
   - Select $y_{ij} = \texttt{Fit}$ with probability $P(y_{ij} = \texttt{Fit}|s_i, t_j, \alpha, b_1, b_2)$ defined in Equation (4).
   - Select $y_{ij} = \texttt{Large}$ with probability $P(y_{ij} = \texttt{Large}|s_i, t_j, \alpha, b_1, b_2)$ defined in Equation (5).

We place Gaussian priors on customer and product true sizes. The product mean size is initialized with the catalog size c_j and the customer mean size μ_i is initialized with the mean of all purchased product sizes that fit the customer i. We introduce Gaussian priors for $b_1 \sim \mathcal{N}(\mu_{b_1}, \sigma_{b_1}^2)$ and $b_2 \sim \mathcal{N}(\mu_{b_2}, \sigma_{b_2}^2)$ to control the movement of the model parameters.

4.3 Posterior Inference for Model Training

The goal of model training is to learn the posterior distribution of latent sizes $P(\beta|D)$. In this section, we restrict our discussion to inference with the Logit link function defined in Equation (2), while we explore variational inference with Probit in the Appendix. To simplify notation, we represent the collective data and labels of all the logistic functions in the likelihood terms in Equations (3), (4) and (5) by $\mathbf{X}$ and $\mathbf{Y}$, respectively. We denote by $\mathbf{D}$ the set of all examples (x_i, y_i) corresponding to transactions in D. Thus, $\mathbf{D}$ contains the example $(x_{ijs}, 1)$ for the transaction $(i, j, \texttt{Small}) \in D$, the examples $(x_{ijs}, 0)$ and $(x_{ijf}, 1)$ for the transaction $(i, j, \texttt{Fit}) \in D$, and the examples $(x_{ijs}, 0)$ and $(x_{ijf}, 0)$ for the transaction $(i, j, \texttt{Large}) \in D$. The joint data distribution can be expressed as:

$$P(\beta, \mathbf{D}|\Sigma) = \left(\prod_{(x,y) \in \mathbf{D}} \sigma(y, \beta^T \cdot x) \cdot \prod_i \mathcal{N}(s_i|\mu_i, \sigma_s^2) \cdot \prod_j \mathcal{N}(t_j|c_j, \sigma_t^2) \cdot \prod_{b \in \{b_1, b_2\}} \mathcal{N}(b|\mu_b, \sigma_b^2) \right) \tag{6}$$

where, $\Sigma = \{\{\mu_i\}, \{c_j\}, \sigma_s, \sigma_t, \mu_{b_1}, \mu_{b_2}, \sigma_{b_1}, \sigma_{b_2}\}$ and $\sigma(y, \theta)$ is the sigmoid function defined in Equation (2). The posterior distribution of β is given by:

$$P(\beta|\mathbf{D}, \Sigma) = \frac{P(\beta, \mathbf{D}|\Sigma)}{\int_\beta P(\beta, \mathbf{D}|\Sigma) d\beta} \propto P(\beta, \mathbf{D}|\Sigma) \tag{7}$$

Note that the posterior distribution in Equation (7) above cannot be computed in closed form due to the presence of logistic likelihood terms and normal priors. In order to make the posterior tractable, we introduce a Polya-Gamma latent variable w for every $(x, y) \in \mathbf{D}$ that generates a logistic term in the likelihood as suggested in [25].

We first define the Polya-Gamma random variable: $X \sim PG(b, a)$ with $b > 0$ and $a \in R$, if

$$X \overset{D}{=} \frac{1}{2\pi^2} \sum_{k=1}^{\infty} \frac{g_k}{(k - 1/2)^2 + a^2/(4\pi^2)} \tag{8}$$

where $\overset{D}{=}$ indicates equality in distribution and $g_k \sim Ga(b, 1)$ are independent gamma random variables.

Let $w \sim PG(1, 0)$. We define the joint likelihood distribution $P(w, y|x, \beta) = \frac{1}{2} e^{((y-1/2)\cdot(\beta^T \cdot x) - w\cdot(\beta^T \cdot x)^2/2)} P(w)$. It can be shown (see Theorem 1 in [25]) that the expectation of w on the joint likelihood distribution is logistic, that is,

$$\begin{aligned} P(y|\beta, X) &= \frac{e^{y\beta^T \cdot x}}{1 + e^{\beta^T \cdot x}} \\ &= \int_0^\infty \frac{1}{2} e^{((y-1/2)\cdot(\beta^T \cdot x) - w(\beta^T \cdot x)^2/2)} P(w) dw \end{aligned} \tag{9}$$

Let $\mathbf{W}$ be the set of Polya-Gamma variables w for $(x, y) \in \mathbf{D}$. Then, we have that $\int P(\mathbf{W}, \beta|\mathbf{D}, \Sigma) d\mathbf{W} \propto \int P(\mathbf{W}, \beta, \mathbf{D}|\Sigma) d\mathbf{W} \propto P(\beta, \mathbf{D}|\Sigma) \propto P(\beta|\mathbf{D}, \Sigma)$. In the following subsections, we present Gibbs sampling and variational inference procedures to approximate the augmented posterior $P(\mathbf{W}, \beta|\mathbf{D}, \Sigma)$ – these give us approximations of β's posterior.

4.3.1 Gibbs Sampling-Based Scheme. Our Gibbs sampling procedure cycles through the w_i and β_j variables, drawing samples for each one from its distribution conditioned on values of the remaining variables. The conditional distribution of w_i is given by (see [25] for details):

$$\begin{aligned} P(w_i|\beta, x_i) &\propto e^{-w_i(\beta^T \cdot x_i)^2/2} P(w_i) \\ &= PG(w_i|1, \beta^T \cdot x_i) \end{aligned} \tag{10}$$

The conditional distribution of β_j is given by

$$\begin{aligned} &P(\beta_j|\beta_{-j}, \mathbf{W}, \mathbf{D}) \\ &\quad = \frac{P(\mathbf{W}, \mathbf{D}|\beta) P(\beta)}{\int_{\beta_j} P(\mathbf{W}, \mathbf{D}|\beta) P(\beta)} d\beta_j \\ &\quad \propto \prod_{(x_i, y_i) \in \mathbf{D}} \prod_{x_{ij} \neq 0} \Big(e^{((y_i - 1/2)(\beta^T \cdot x_i) - w_i \cdot (\beta^T \cdot x_i)^2/2)} . \\ &\qquad\qquad \mathcal{N}(\beta_j|\mu_{\beta_j}, \sigma_{\beta_j}^2) \Big) \\ &\quad = \mathcal{N}(\beta_j|m_j, V_j) \end{aligned} \tag{11}$$

where,

$$\frac{1}{V_j} = \frac{1}{\sigma_{\beta_j}^2} + \sum_{(x_i, y_i) \in \mathbf{D}} \sum_{x_{ij} \neq 0} w_i \cdot x_{ij}^2$$

and

$$m_j = V_j \left(\frac{\mu_{\beta_j}}{\sigma_{\beta_j}^2} + \sum_{(x_i, y_i) \in \mathbf{D}} \sum_{x_{ij} \neq 0} \Big((y_i - 1/2) \cdot x_{ij} - w_i \cdot x_{ij} \cdot \sum_{l \neq j} \beta_l \cdot x_{il} \Big) \right)$$

4.3.2 Variational Inference-Based Scheme. We apply mean-field variational inference to make the training process efficient. We approximate the posterior distribution using a proposal distribution $q(\mathbf{W}, \beta)$ that factorizes as follows:

$$q(\mathbf{W}, \beta) = \prod_i q(w_i) . \prod_j q(\beta_j)$$

The best q^* distribution is one that minimizes the Kullback-Leibler divergence $KL(q(\mathbf{W}, \beta)||P(\mathbf{W}, \beta|\mathbf{D}))$. For $q(w_i)$, the KL divergence is minimum when

$$\log q^*(w_i) = E_{-w_i}[\log P(w_i|\beta, x_i)] + const$$

Substituting $P(w_i|\beta, x_i) = PG(w_i|1, \beta^T x_i)$ from Equation (10), in the above equation, we get:

$$\begin{aligned} \log q^*(w_i) &= E_\beta[\log P(w_i|\beta, x_i)] \\ &= E_\beta[-w_i(\beta^T \cdot x_i)^2/2 + \log P(w_i)] + const \\ &= -w_i \cdot \Big[\Big(\sum_j E[\beta_j] \cdot x_{ij} \Big)^2 + \sum_j Var[\beta_j] \cdot x_{ij}^2 \Big] / 2 \\ &\qquad + \log P(w_i) + const \end{aligned}$$

From the above expression, it can be seen that $q^*(w_i)$ is $PG(1, a_i)$, where

$$a_i = \sqrt{\Big(\sum_j E[\beta_j] \cdot x_{ij}\Big)^2 + \sum_j Var[\beta_j] \cdot x_{ij}^2} \tag{12}$$

Similarly, for β_j, the KL divergence is minimum when

$$\log q^*(\beta_j) \propto E_{-\beta_j}[\log P(\beta_j|\beta_{-j}, \mathbf{W}, \mathbf{D})] + const \tag{13}$$

Substituting for $P(\beta_j|\beta_{-j}, \mathbf{W}, \mathbf{D})$ from Equation (11), Equation (13) simplifies to the following:

$$q^*(\beta_j) = \mathcal{N}(\hat{m_j}, \hat{V_j})$$

where,

$$\frac{1}{\hat{V_j}} = \left(\frac{1}{\sigma_{\beta_j}^2} + \sum_{(x_i, y_i) \in \mathbf{D} \wedge x_{ij} \neq 0} E[w_i] \cdot x_{ij}^2\right) \tag{14}$$

$$\hat{m_j} = \hat{V_j}\left(\frac{\mu_{\beta_j}}{\sigma_{\beta_j}^2} + \sum_{(x_i, y_i) \in \mathbf{D} \wedge x_{ij} \neq 0} \Big((y_i - 1/2) \cdot x_{ij} - E[w_i] \cdot x_{ij} \cdot \Big(\sum_{l \neq j} E[\beta_l] \cdot x_{il}\Big)\Big)\right) \tag{15}$$

For $q^*(w_i)$ defined in Equation (12), $E[w_i] = \frac{1}{2a_i}\tanh(a_i/2)$, and for $q^*(\beta_j)$ defined in Equation (4.3.2), $E[\beta_j] = \hat{m_j}$ and $Var[\beta_j] = \hat{V_j}$. Starting with random initial values, we iteratively update $E[w_i]$, $E[\beta_j]$ and $Var[\beta_j]$ until convergence. These are used to compute parameter values a_i, $\hat{V_j}$ and $\hat{m_j}$ as shown in Equations (12), (14) and (15).

A Note on the computational complexity: Equation (12) involves a constant time operation per transaction, owing to the sparse nature of data vectors x_{ij} leading to $O(|\mathbf{D}|)$ complexity for computing the updates for w variables. Computing the updates for all β_j variables leads to $O(|\mathbf{D}|)$ complexity since Equation (14) and (15) involve each transaction contributing twice while updating β_j variables, once for the corresponding customer and once for the product. Hence, our inference technique is efficient and scales linearly $O(|\mathbf{D}|)$ with the data.

4.4 Recommending Product Sizes

Given a customer i, we recommend the product j whose predictive distribution for size fit probability is maximum. Let $\beta^1 = (s_1^1, \ldots, s_c^1, t_1^1, \ldots, t_p^1, b_1^1, b_2^1), \ldots \beta^r = (s_1^r, \ldots, s_c^r, t_1^r, \ldots, t_p^r, b_1^r, b_2^r)$ be the r samples drawn from the posterior of β using the Gibbs sampling or variational inference procedures described in Sections 4.3.1 and 4.3.2, respectively. The predictive distribution for fit size probability is obtained by marginalizing with respect to the posterior distribution of β.

$$\begin{aligned} P(y_{ij} = \mathtt{Fit}|\mathbf{D}, \Sigma) &= \int_\beta P(y_{ij} = \mathtt{Fit}|\beta, \Sigma) \cdot P(\beta|\mathbf{D}, \Sigma) d\beta \\ &= \int_\beta \Big(P(y_{ijs} = 0|\beta^T \cdot x_{ijs}) \cdot P(y_{ijf} = 1|\beta^T \cdot x_{ijf}) \cdot \\ &\qquad P(\beta|\mathbf{D}, \Sigma)\Big) d\beta \\ &\approx \frac{1}{r}\sum_{l=1}^{r} \frac{1}{1 + e^{\beta^{l^T} \cdot x_{ijs}}} \cdot \frac{e^{\beta^{l^T} \cdot x_{ijf}}}{1 + e^{\beta^{l^T} \cdot x_{ijf}}} \end{aligned} \tag{16}$$

Thus, the fit probability score $P(y_{ij} = \mathtt{Fit}|\mathbf{D}, \Sigma)$ is obtained by simply averaging the fit probability values for the r samples. We can also use the samples $\beta^1, \ldots \beta^r$ to compute confidence intervals for the fit probability score. Let $p_l = \sigma(0, \beta^{l^T} \cdot x_{ijs}) \cdot \sigma(1, \beta^{l^T} \cdot x_{ijf})$ be the fit probability score obtained from the l^{th} sample. Then, we can take different percentile values of the set of probability values $p_1, \ldots, p_r$ to obtain the lower and upper boundaries of the confidence interval for the fit probability score. For instance, the 5^{th} and 95^{th} percentile values will give us the 90% confidence interval for the fit probability score. Note that for customers and products in a small number of transactions, the size posterior distributions will have high variance. Thus, the confidence intervals for such customers and products will be large. In contrast, customers and products involved in many transactions will have size posterior distributions with low variance and narrow confidence intervals.

Once we have the fit probability score and the confidence interval for the probability score for each product (child of parent product), one option is to recommend the product with the maximum fit probability score. Another alternative is to factor in the uncertainty in fit probability score estimates, and recommend the product with the highest lower bound of fit probability score confidence intervals.

5 EXPERIMENTS

In this section, we evaluate our Bayesian models for size recommendation on real-world Amazon shoes data and synthetically generated datasets. We first describe our experimental setup detailing our evaluation metrics and baselines in Section 5.1, followed by experiments with Amazon datasets in Section 5.2 and synthetic datasets in Section 5.3.

5.1 Experimental Setup

We use offline shoes datasets containing customer transactions for evaluating model performance. Such an offline setup is restrictive, since we do not observe (1) customer's response when our recommendation is shown (2) the true latent size of the customer. The only available data in an offline setting is the transaction carried out by the customer and the resulting *fitSuitabilityCode* of the transaction. Hence, we evaluate the performance of different size recommendation schemes by the ability of the classifier, constructed based on the learnt latent customer and product sizes from our model, to predict the *fitSuitabilityCode* outcome on unseen transactions. Note that a similar setup is also used in [30] for carrying out model performance benchmarking.

We carry out a time-based split of available customer transactions into training and test set. We compare the results of our Bayesian models on held out test transactions against a baseline model with (1) catalog size as a latent size of products and (2) customer's latent size is the mean of all catalog sizes purchased by the customer with a *fitSuitabilityCode* of `Fit`. The baseline algorithm also learns customer and product true sizes on the training set. The learnt customer and product latent sizes are used to train a Logistic Regression (Linear) and Random Forest (RF) classifiers to produce *fitSuitabilityCode* as output. Our Bayesian models are compared

A	B	C	D	E	F
10.4M	17M	33.2M	25M	12.9M	27M

Table 1: Summary of transaction count for all datasets.

against the performance metric of the baseline Linear and RF classifiers on the test set. For the Bayesian logit ('Logit') and probit ('Probit') schemes, we compute the probability scores on the test set using Equation (16), and then predict the *fitSuitabilityCode* outcome for a transaction as the one with the highest probability. We also compare our technique against the state-of-the-art technique of [30] that learns customer and product sizes through coordinate descent on a loss function constructed from customer purchase and return history.

5.2 Experiments with Shoes Data

Data Analysis:

Our dataset comprises customer transactions: *customerId*, *productId* and *fitSuitabilityCode* ∈ {`Small`, `Fit`, `Large`}. Each transaction is further categorized based on product category and assigned a label A-F due to proprietary reasons. Table 1 represents the transaction counts for all product categories. For each product category, we partition the transactions in temporal order with first 80% in the training set and the remaining 20% in the test set.

The catalog size of products in our datasets represents the US sizing scheme with sizes varying between 4 and 20 at 0.5 intervals. The histogram of products and transactions for different catalog sizes is represented in Figures 2 and 3 respectively. The actual count of products and transactions are not reported to safeguard proprietary information. Figure 2 and 3 indicates that customers prefer buying full catalog sizes over half catalog sizes, perhaps due to lack of availability of half-sizes in the catalog for many products. Also, note that except for category "C", majority of purchased products have catalog sizes between 7 and 10, while for category "C" majority of purchased products have catalog sizes between 8 and 12. Note that there are very few purchases for products with catalog sizes greater than 14, with almost negligible count of transactions for catalog sizes more than 17.

Figures 4 and 5 indicates the histogram of products and customers, respectively with different transactions counts. Once again, the actual numbers in the figures are not reported to safeguard propriety information. Note that very few products and customers have more than 3 transactions in the training data, indicating a sparse purchase graph.

Experimental Results: The results of our offline experiments with various datasets are shown in Table 2 ('Logit' refers to the Variational Bayesian model with logit link function and 'Probit' refers to the Variational Bayesian model with probit link function). For proprietary reasons, we do not report the absolute weighted AUC numbers for different models, but rather report the percentage improvements in weighted AUC scores over and above the Baseline Linear model. We also compare our results with those obtained using work carried out by [30]. Observe that our Bayesian models show a relative improvement of 17-26% in weighted AUC over baseline model. Our models also outperform models built by [30] on 5 out of the 6 datasets we experimented with, as reported in Table 2. This can be attributed to our Bayesian models following a principled probabilistic approach that allows them to (1) overcome data sparsity by placing priors on latent sizes, (2) capture uncertainty due to noise in the data in the inferred latent size posteriors, and (3) propagate the uncertainty into the final fit probability scores by averaging over the posterior. Note that the logit and probit models perform similarly (except for "B" category), because the logit and probit link functions are very similar.

Dataset	RF	[30]	Logit	Probit
D	2.71%	20.16 %	25.78%	**26.16**%
E	1.48%	15.58 %	**20.22**%	20.04%
F	2.50%	17.31 %	19.42%	**20.00**%
C	3.79%	16.10 %	**19.70**%	19.32%
A	0.55%	17.16%	17.71%	**17.90**%
B	2.05%	**21.27** %	18.28%	20.52%

Table 2: Summary of offline experimental results. We report percentage improvements in weighted AUC over and above the Baseline Linear model for the Logit, Probit, the model built by [30] and a baseline random forest model. (Best results shown first.)

5.3 Experiments with Synthetic Shoes Data

In the real-world setup described in Section 5.2, due to problems associated with observability of latent variables like true sizes and variability in data, we are only able to make indirect inferences on the power of our algorithms in estimating the true size distribution. Hence, in this section, we carry out experiments to measure the ability of our inference procedures to recover true latent sizes for customers and products on simulated datasets.

Data Generation Process: We consider catalog sizes from 4 to 20 in increments of 0.5. We follow the generative process described in Section 4.2 to generate the true size of product j as $t_j \sim \mathcal{N}(c_j, \hat{\sigma}_t^2)$ (where c_j is a randomly chosen catalog size) and the true size $s_i \sim \mathcal{N}(10, 3)$ of customer i is drawn from a normal truncated between 4 and 20. We generate sizes for 10,000 child products and 10,000 customers. To generate a transaction, we pick a customer at random and randomly pick any child product having the catalog size closest to the customer true size. We generate thresholds b_1 and b_2 from a normal prior and categorize a transaction as {`Fit, Small, Large`} based on the generative process described in Section 4.2.

Experimental Results: The RMSE between estimated and simulated latent true sizes for customers and products are shown in Table 3 along with the AUC on size fit prediction for different values of σ_t^2 introduced when simulating product true sizes. We observe that (1) Both the 'Logit' and 'Probit' models perform very well on the size fit prediction with AUC, between 0.886 and 0.972 (2) As can be expected, our methods are able to infer the latent product and customer true sizes more accurately for smaller magnitudes of variance σ_t^2 introduced when simulating product true sizes. We also note that for higher values of variance σ_t^2, the RMSE of product sizes to true sizes is much lower than the introduced variance in generating product sizes. Such a behavior can be attributed to a

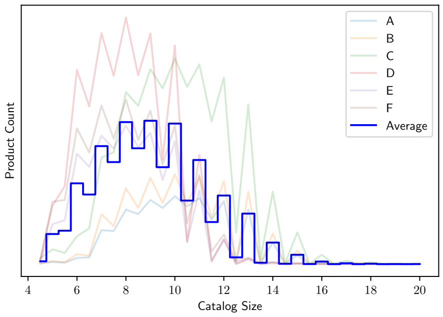


Figure 2: Product count across catalog size.

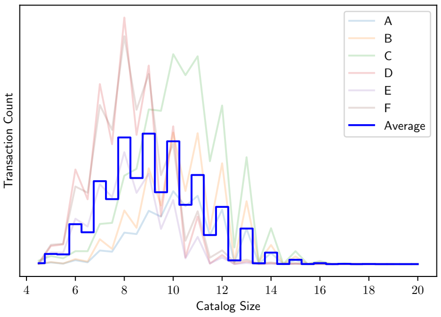


Figure 3: Transaction count across catalog size.

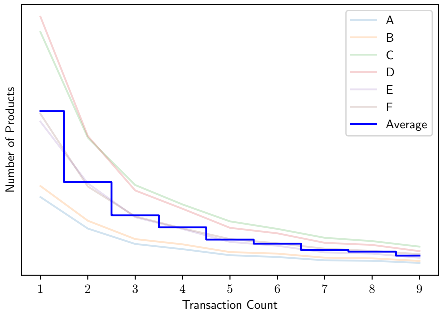


Figure 4: Histogram of products with transaction count.

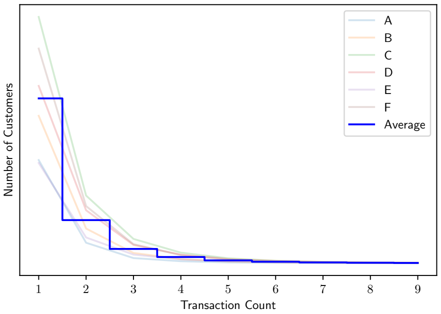


Figure 5: Histogram of customers with transaction count.

Product Variance	Fit %	Cust RMSE	Cust RMSE	Product RMSE	Product RMSE	Weighted AUC	Weighted AUC
		Logit	Probit	Logit	Probit	Logit	Probit
0.5	94.27	0.275	**0.227**	**0.412**	0.413	**0.901**	0.886
0.7	83.61	0.324	**0.305**	**0.454**	0.471	**0.932**	0.928
0.9	72.59	**0.318**	0.330	**0.473**	0.502	**0.956**	0.951
1.1	63.83	0.305	**0.266**	**0.507**	0.564	**0.959**	0.951
1.3	55.90	0.294	**0.269**	**0.582**	0.684	**0.968**	0.961
1.5	49.29	0.286	**0.269**	**0.708**	0.841	**0.972**	0.965

Table 3: Summary of RMSE on synthetic data.

larger percentage of `small` and `large` transactions being available in the simulated data, leading to reduced class imbalance. A similar trend is also observed in terms of the weighted-AUC metric. (3) Like the results obtained on real-world datasets of Section 5.2, the 'Logit' and 'Probit' models perform similarly to each other, with differences being evident only in the third decimal place for the weighted-AUC metric.

6 EXTENSIONS

We describe extensions to our models to (1) address cold start scenarios using customer and product features in Section 6.1, (2) handling purchases for multiple personas in an account in Section 6.2, and (3) leverage multi-dimensional sizes like length and width in Section 6.3.

6.1 Leveraging Customer and Product Features

We can leverage additional customer information such as gender, age and subscription to loyalty programs and product information such as category, brand and product type in addition to catalog size. Leveraging such features can handle cold start scenarios and combat data sparsity thus improving accuracy of our model.

Let the feature vectors of customer i and product j be f_i and g_j, respectively. We can incorporate the features in our model as follows:

(1) Include the feature vectors f_i and g_j in the data vectors x_{ijs} and x_{ijf}, and the corresponding weight vectors w_f and w_g in the latent size vector β.

(2) For each customer i, draw true size $s_i \sim \mathcal{N}(w'_f \cdot f_i, \sigma_s^2)$. For each customer j, draw true size $t_j \sim \mathcal{N}(w'_g \cdot g_j, \sigma_t^2)$. Thus, the means of customer and product true sizes are obtained by performing regression over their features.

The weight values for w_f, w_g, w'_f, w'_g can be computed using a Monte-Carlo EM algorithm as described in [2].

6.2 Handling Personas

In practice, a customer account may be shared by multiple individuals with different sizes, for example, different family members including children and adults. Thus, learning a single true size per customer account may lead to inaccurate true size estimates. We can model multiple personas per customer account by introducing new latent variables z_{ij} for each transaction $(i, j, y_{ij}) \in D$. The latent variables z_{ij} capture the persona of customer i that purchases product j. We associate a separate latent size s_{ik} with the k^{th} persona of customer i, and the sizes for all customer personas are included in the latent size vector β. Furthermore, the data vectors x_{ijs} and x_{ijf} have values α and $-\alpha$ present in the positions of $s_{iz_{ij}}$ and t_j, respectively. Our generative model (described in Section 4.2) is extended as follows. First, we draw a persona distribution θ_i for each customer i from a Dirichlet distribution. For each transaction (i, j, y_{ij}), we first draw the persona z_{ij} from multinomial ($Mult(\theta_i)$) and then draw the transaction outcome with probabilities given by the likelihood functions defined in Equations (3)-(5). Our Gibbs sampling and variational inference procedures can be easily extended to approximate the posterior distributions of latent variables β and $\{z_{ij}\}$.

6.3 Multi-Dimensional Sizes

We can extend our techniques to infer multi-dimensional true size vectors for customers and products. The various dimensions capture different aspects related to size such as length and width for shoes, or waist and length for jeans. Let d be the number of size dimensions. Then the customer true sizes s_i and product true sizes t_j (in latent size vector β) as well as the scale parameter α (in data vectors x_{ijs} and x_{ijf}) are d-dimensional vectors.

7 CONCLUSIONS

In this paper, we proposed novel Bayesian logit and probit regression models with ordinal categories to model size fits, and efficient algorithms for posterior inference based on mean-field variational inference and Polya-Gamma augmentation. In experiments with real-life shoe datasets, our Bayesian models delivered 17-26% higher AUCs compared to baselines when predicting size fit outcomes of customer purchase transactions. Our models also outperform state-of-the-art baselines in 5 of 6 real-world datasets. On simulated datasets, we show that our models are able to reduce significant proportion of introduced size variances. In terms of a choice between using either a logit or a probit link function for our Bayesion model, we find that the two models perform very similarly to each other on both real-world and simulated datasets. They also have similar inference times during the model training process (~ 1 minute per iteration), thereby not leaving us with an explicit winner between the two.

ACKNOWLEDGEMENT

We would like to thank Dr. Matthias Seeger for his valuable comments on an earlier version of the manuscript.

REFERENCES

[1] Adomavicius, G., and Tuzhilin, A. Context-aware recommender systems. In *Recommender systems handbook*. Springer, 2011, pp. 217–253.

[2] Agarwal, D., and Chen, B.-C. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2009), KDD '09, ACM, pp. 19–28.

[3] Aggarwal, C. C. *Recommender systems*. Springer, 2016.

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

[5] Bauer, J., and Nanopoulos, A. A framework for matrix factorization based on general distributions. In *Proceedings of the 8th ACM Conference on Recommender Systems* (New York, NY, USA, 2014), RecSys '14, ACM, pp. 249–256.

[6] Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.

[7] Charlin, L., Ranganath, R., McInerney, J., and Blei, D. M. Dynamic poisson factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems* (New York, NY, USA, 2015), RecSys '15, ACM, pp. 155–162.

[8] Chen, M.-H., and Dey, D. K. Bayesian analysis for correlated ordinal data models. *BIOSTATISTICS-BASEL- 5* (2000), 133–158.

[9] Chu, W., and Park, S.-T. Personalized recommendation on dynamic content using predictive bilinear models. In *Proceedings of the 18th International Conference on World Wide Web* (New York, NY, USA, 2009), WWW '09, ACM, pp. 691–700.

[10] Church, K., Smyth, B., Cotter, P., and Bradley, K. Mobile information access: A study of emerging search behavior on the mobile internet. *ACM Trans. Web 1*, 1 (May 2007).

[11] Gopalan, P. K., Charlin, L., and Blei, D. Content-based recommendations with poisson factorization. In *Advances in Neural Information Processing Systems* (2014), pp. 3176–3184.

[12] Greene, W. H., and Hensher, D. A. *Modeling ordered choices: A primer*. Cambridge University Press, 2010.

[13] Grimmer, J. An introduction to bayesian inference via variational approximations. *Political Analysis 19*, 1 (2010), 32–47.

[14] Harvey, M., Carman, M. J., Ruthven, I., and Crestani, F. Bayesian latent variable models for collaborative item rating prediction. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management* (New York, NY, USA, 2011), CIKM '11, ACM, pp. 699–708.

[15] Joshi, B., Iutzeler, F., and Amini, M.-R. Asynchronous distributed matrix factorization with similar user and item based regularization. In *Proceedings of the 10th ACM Conference on Recommender Systems* (New York, NY, USA, 2016), RecSys '16, ACM, pp. 75–78.

[16] Karatzoglou, A., Amatriain, X., Baltrunas, L., and Oliver, N. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems* (2010), ACM, pp. 79–86.

[17] Kawakatsu, H., and Largey, A. G. EM algorithms for ordered probit models with endogenous regressors. *The Econometrics Journal 12*, 1 (2009), 164–186.

[18] Koren, Y., Bell, R., and Volinsky, C. Matrix factorization techniques for recommender systems. *Computer 42*, 8 (2009).

[19] Liang, D., Altosaar, J., Charlin, L., and Blei, D. M. Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In *Proceedings of the 10th ACM conference on recommender systems* (2016), ACM, pp. 59–66.

[20] McAuley, J., and Leskovec, J. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems* (2013), ACM, pp. 165–172.

[21] McKinley, T. J., Morters, M., Wood, J. L., et al. Bayesian model choice in cumulative link ordinal regression models. *Bayesian Analysis 10*, 1 (2015), 1–30.

[22] Narita, A., Hayashi, K., Tomioka, R., and Kashima, H. Tensor factorization using auxiliary information. *Data Mining and Knowledge Discovery 25*, 2 (2012), 298–324.

[23] Palmisano, C., Tuzhilin, A., and Gorgoglione, M. Using context to improve predictive modeling of customers in personalization applications. *IEEE transactions on knowledge and data engineering 20*, 11 (2008), 1535–1549.

[24] Panniello, U., Tuzhilin, A., Gorgoglione, M., Palmisano, C., and Pedone, A. Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems. In *Proceedings of the third ACM conference on Recommender systems* (2009), ACM, pp. 265–268.

[25] Polson, N. G., Scott, J. G., and Windle, J. Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American statistical Association 108*, 504 (2013), 1339–1349.

[26] Qin, Z., Rishabh, I., and Carnahan, J. A scalable approach for periodical

personalized recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems* (New York, NY, USA, 2016), RecSys '16, ACM, pp. 23–26.

[27] Rafailidis, D., and Nanopoulos, A. Modeling the dynamics of user preferences in coupled tensor factorization. In *Proceedings of the 8th ACM Conference on Recommender Systems* (New York, NY, USA, 2014), RecSys '14, ACM, pp. 321–324.

[28] Salakhutdinov, R., and Mnih, A. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th International Conference on Machine Learning* (New York, NY, USA, 2008), ICML '08, ACM, pp. 880–887.

[29] Salakhutdinov, R., and Mnih, A. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems* (2008), vol. 20.

[30] Sembium, V., Rastogi, R., Saroop, A., and Merugu, S. Recommending product sizes to customers. In *Proceedings of the Eleventh ACM Conference on Recommender Systems* (New York, NY, USA, 2017), RecSys '17, ACM, pp. 243–250.

[31] Vasile, F., Smirnova, E., and Conneau, A. Meta-prod2vec: Product embeddings using side-information for recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems* (2016), ACM, pp. 225–232.

A APPENDIX: MODELING WITH PROBIT

In this section, we discuss inference for our model with the Probit link function. Consider data $\{x_i, y_i\}$, where $\{y_i \in \{0, 1\}\}$ are the observed category labels and $\{x_i \in R^M\}$ the features. Let $\beta \in R^M$. The binary probit is defined as follows where Φ is the CDF of the normal distribution with mean 0 and variance σ_p^2:

$$p(y_i = 1|x_i) = \Phi(x_i^T \beta; \sigma_p^2)$$

Since the CDF of the normal is not a tractable function[4, 17], an augmentation strategy was introduced by [4] for bayesian inference where the probit is defined in terms of the following linear model in terms of an auxilliary random variable z_i:

$$\begin{aligned} z_i &= \beta^t x_i + \epsilon,\ \epsilon \in \mathcal{N}(0, \sigma_p^2) \qquad (17) \\ y_i &= 1 \text{ if } z_i > 0 \\ &= 0 \text{ otherwise} \end{aligned}$$

A.1 Variational Inference with Probit for Size Recommendation

We perform approximate inference using variational inference[6] with the mean field approximation. Let $\beta = \{\{s_i\}, \{t_j\}, b_1, b_2\}$, the set of all latent variables in the model. Consider a set Z containing the auxiliary variables for all the probits defined in $\mathbf{D}$, that is defined in Section 4.3. The joint log likelihood $p(\beta, Z, \mathbf{D})$ can be expanded as follows:

$$\begin{aligned} \log\ p(\beta, Z, \mathbf{D}) = &\sum_{\beta_j} \log\ p(\beta_j) \\ &+ \sum_{\{x_i, y_i\} \in \mathbf{D}, y_i = 0} \log\left(\mathcal{N}(z_i, \beta^T x_i, \sigma_p^2) \cdot \delta(z_i > 0)\right) \\ &+ \sum_{\{x_i, y_i\} \in \mathbf{D}, y_i = 1} \log\left(\mathcal{N}(z_i, \beta^T x_i, \sigma_p^2) \cdot \delta(z_i < 0)\right) \end{aligned}$$

We note that since we are working with the joint likelihood $p(\beta, Y)$, we deal with terms containing $p(z_i, y_i|\beta)$ that has a simpler form than the truncated Gaussian $p(z_i|y_i, \beta)$ which has a not so tractable normalization constant. With this, we now derive the variational updates for the latent variables in our model.

Update for $\{z_i\}$:

$$\begin{aligned} \log\ q^*(z_i) &= E_{-z^i}[\log\ p(\beta, Z, \mathbf{D})] + const \\ &= E_{-z_i}[\log\ p(z_i, y_i|\beta)] + const \\ &= E_{-z_i}[\log\left(\delta(y_i, z_i > 0)\ \cdot \mathcal{N}(z_i; \beta^T x_i, \sigma_p^2)\right)] + const \\ &= E_{-z_i}[\log\left(\delta(y_i, 1) \cdot \mathcal{N}^+(z_i; \beta^T x_i, \sigma_p^2)\right. \\ &\qquad \left. + \delta(y_i, 0) \cdot \mathcal{N}^-(z_i; \beta^T x_i, \sigma_p^2)\right)] + const \end{aligned}$$

This takes the form of a truncated Gaussian. Hence, we obtain $q^*(z_i)$ as a truncated Gaussian and $E^*[z_i]$ the expectation of a truncated Gaussian[13] :

$$\begin{aligned} q^*(z_i) &= \mathcal{N}^-(z_i, E^*[\beta]^T x_i, \sigma_p^2)^{\delta(y_i, 0)} \cdot \mathcal{N}^+(z_i, E^*[\beta]^T x_i, \sigma_p^2)^{\delta(y_i, 1)} \\ E^*[z_i] &= \delta(y_i, 0) \cdot \left(E^*[\beta]^T x_i - \sigma_p \frac{\phi_i}{\Phi_i}\right) \\ &\qquad + \delta(y_i, 1) \cdot \left(E^*[\beta]^T x_i + \sigma_p \frac{\phi_i}{1 - \Phi_i}\right) \qquad (18) \end{aligned}$$

Where $E^*[\beta]$ is computed in later steps from $q^*(\beta)$. Note that ϕ_i and Φ_i are the pdf and the CDF of the standard normal evaluated at $\frac{E^*[\beta]^T x_i}{\sigma_p}$.

Update for $\{\beta_j\}$:

$$\log\ q^*(\beta_j) = E_{-\beta_j}[\log\ p(\beta, Z, \mathbf{D})] \qquad (19)$$

Consolidating all the terms with β_j from Equation 18:

$$\begin{aligned} \log\ q^*(\beta_j) &= E_{-\beta_j}[\log\ p(\{\beta_j\}) + \log\ p(\{Z, \mathbf{D}|\beta)] + const \\ &= E_{-\beta_j}[\log\ p(\beta_j; \mu_{\beta_j}, \sigma_{\beta_j}^2) \prod_{\{x_i, y_i\} \in \mathbf{D}} p(y_i) p(z_i|\beta, y_i)] \\ &= -\frac{(\beta_j - \mu_{\beta_j})^2}{2\sigma_{\beta_j}^2} - \frac{1}{2} \sum_{\{x_i, y_i\} \in \mathbf{D}} E_{-\beta_j}[(z_i - \sum_{k \neq j} \beta_k^T x_{ik})^2] \\ &= \mathcal{N}(\frac{b_{\beta_j}}{a_{\beta_j}}, \frac{1}{a_{\beta_j}}) \qquad (20) \end{aligned}$$

$$\begin{aligned} a_{\beta_j} &= \frac{1}{\sigma_{\beta_j}^2} + |\mathbf{D}| \\ b_{\beta_j} &= \frac{\mu_{\beta_j}}{\sigma_{\beta_j}^2} + \sum_{\{x_i, y_i\} \in \mathbf{D}} \left(E^*[z_i] - \sum_{k \neq j} E^*[\beta_k]\, x_{ik}\right) \qquad (21) \end{aligned}$$

$$E^*[\beta_j] = \frac{\frac{\mu_{\beta_j}}{\sigma_{\beta_j}^2} + \sum_{\{x_i, y_i\} \in \mathbf{D}} (E^*[z_i] - \sum_{k \neq j} E^*[\beta_k]\, x_{ik})}{\frac{1}{\sigma_{s_i}^2} + |\mathbf{D}|}$$

where $|\mathbf{D}|$ is the cardinality of set $\mathbf{D}$.