Submitted to *INFORMS Journal on Computing*
manuscript JOC-2023-10-OA-0372

Supplementary Material for: An Interpretable Preference Learning Model Admitting Dynamic and Context-Dependent Preferences

---

The supplementary material includes the methods for posterior inference and prediction based on the expected comprehensive value of each item (Appendix A), experimental results on the Google Maps and MovieLens datasets (Appendix B), descriptions of benchmark models used for predictive evaluation (Appendix C), and robustness analyses conducted on datasets with varying characteristics (Appendix D). Due to page limitations, Appendix B is provided in the SSRN version at `https://ssrn.com/abstract=4606645`.

Appendix A. Posterior Inference and Prediction

This section aims to infer the estimated values of latent parameters and the posterior distributions of the latent variables in our model and generate predictive recommendations for items with undisclosed user preferences. On the one hand, we propose a dedicated deep variational inference method to approximate the latent parameters in our model (see Section A.1). On the other hand, utilizing the estimated parameters, we compute the posterior expectation of the comprehensive value for each unrated item, which serves as a predictive recommendation for the user (see Section A.2).

A.1. Deep Variational Inference

The complete posterior distribution of the latent variables in our model is as follows:

$$
\begin{aligned}
&p\left(\{\boldsymbol{\gamma}_j^+\},\{\boldsymbol{\gamma}_j^-\},\{y_{j,s}^+\},\{y_{j,s}^-\}\mid\{w_{j,s}^+\},\{w_{j,s}^-\},\{\boldsymbol{\phi}_k^+\},\{\boldsymbol{\phi}_k^-\},\mathcal{H}_1\right)\cdot p\left(\{\boldsymbol{\rho}_{i,t}\},\{z_{i,t,n}\}\mid\{x_{i,t,n}\},\{r_{i,t,n}\},\{\boldsymbol{\beta}_m\},\{\mathbf{u}_{i,m}\},\mathcal{H}_2\right)\\
&\propto\prod_{j=1}^{\mathcal{J}}\prod_{s=1}^{\mathcal{S}_j^+}p(w_{j,s}^+\mid\boldsymbol{\phi}^+_{y_{j,s}^+})p(y_{j,s}^+\mid\boldsymbol{\gamma}_j^+)\times\prod_{j=1}^{\mathcal{J}}\prod_{s=1}^{\mathcal{S}_j^-}p(w_{j,s}^-\mid\boldsymbol{\phi}^-_{y_{j,s}^-})p(y_{j,s}^-\mid\boldsymbol{\gamma}_j^-)\times\prod_{j=1}^{\mathcal{J}}p(\boldsymbol{\gamma}_j^+\mid\boldsymbol{\varphi}^+)p(\boldsymbol{\gamma}_j^-\mid\boldsymbol{\varphi}^-)\\
&\quad\times\prod_{i=1}^{\mathcal{I}}\prod_{t=1}^{\mathcal{T}}p(\boldsymbol{\rho}_{i,t}\mid\boldsymbol{\rho}_{i,t-1},\Sigma_\rho)\times\prod_{i=1}^{\mathcal{I}}\prod_{t=1}^{\mathcal{T}}\prod_{n=1}^{\mathcal{N}_{i,t}}p(r_{i,t,n}\mid\mathbf{u}_{i,z_{i,t,n}}^{\mathrm{T}}\hat{\boldsymbol{\tau}}_{x_{i,t,n}},\sigma^2)p(x_{i,t,n}\mid\boldsymbol{\beta}_{z_{i,t,n}})p(z_{i,t,n}\mid\boldsymbol{\rho}_{i,t}),
\end{aligned}
\tag{A.1}
$$

where $\mathcal{H}_1$ and $\mathcal{H}_2$ represent the sets of prior parameters in our model. Like many Bayesian models, the normalization term's intractability prevents the availability of closed-form expressions for the posterior distribution of model parameters (Blei et al. 2003). Consequently, it is necessary to employ approximate methods for estimating the posterior distribution. The Markov Chain Monte Carlo (MCMC) method is a prevalent approach in Bayesian inference, employed to sample from the posterior distribution through the construction of a Markov chain that targets the desired distribution. Despite its broad application, MCMC suffers from slow convergence, requires meticulous tuning of the proposal distribution, and necessitates extensive sampling, which collectively make it computationally intensive, particularly for complex, high-dimensional models. In response to these challenges, Variational Inference (VI) has emerged as a promising alternative, which employs a deterministic optimization technique to approximate the posterior distribution by minimizing Kullback and Leibler (1951) (KL) divergence between the true posterior and its approximation. In contrast to sampling methods that rely on random sampling, VI directly approximates the posterior distribution by iteratively updating the parameters of the variational distribution. This approach achieves computational efficiency and scalability, making it well-suited for large datasets and high-dimensional models. However, traditional VI primarily relies on mean-field theory and employs simple variational distributions to approximate the target posterior distribution (Murphy 2012). While efficient, the simplicity of these distributions restricts the model's fitting capacity and fails to capture the complex interactions among variables in the true posterior. To tackle this problem, we introduce a deep VI method, inspired by Srivastava and Sutton (2016), to estimate the values and posteriors of the latent components in our model. It is important to emphasize that such a deep method does not compromise our model's interpretability, as the neural network involved is dedicated solely to approximating the true posterior distribution in the inference part. In contrast, our model's generative process, which is the basis of its interpretability, remains free of any black-box components that might reduce transparency.

Let $\mathcal{Z}$ and $\mathcal{X}$ denote the sets of latent variables and other components (including observed data, priors, and hyper-parameters), respectively, in our model. Then, the joint distribution can be represented by $p(\mathcal{X},\mathcal{Z})$, while the posterior distribution of latent variables can be represented by $p(\mathcal{Z}\mid\mathcal{X})$. The core concept of our deep VI is to approximate the intractable posterior distribution $p(\mathcal{Z}\mid\mathcal{X})$ with the variational network $q(\mathcal{Z}\mid\mathcal{X})$, where $q(\mathcal{Z}\mid\mathcal{X})$ is a neural network that takes the observed data $\mathcal{X}$ as input and outputs the estimated posteriors of the latent variables $\mathcal{Z}$. Notably, in the Bayesian model, the following formula always holds:

$$\log p(\mathcal{X}) = \int q(\mathcal{Z}\mid\mathcal{X})\log p(\mathcal{X},\mathcal{Z})\mathrm{d}\mathcal{Z} - \int q(\mathcal{Z}\mid\mathcal{X})\log q(\mathcal{Z}\mid\mathcal{X})\mathrm{d}\mathcal{Z} - \int q(\mathcal{Z}\mid\mathcal{X})\log\frac{p(\mathcal{Z}\mid\mathcal{X})}{q(\mathcal{Z}\mid\mathcal{X})}\mathrm{d}\mathcal{Z}. \tag{A.2}$$

The final term $-\int q(\mathcal{Z}\mid\mathcal{X})\log\frac{p(\mathcal{Z}\mid\mathcal{X})}{q(\mathcal{Z}\mid\mathcal{X})}\mathrm{d}\mathcal{Z} > 0$ represents the KL divergence, a measure of dissimilarity between the variational network $q(\mathcal{Z}\mid\mathcal{X})$ and the posterior distribution $p(\mathcal{Z}\mid\mathcal{X})$. This divergence is equal to zero if and only if $q(\mathcal{Z}\mid\mathcal{X}) = p(\mathcal{Z}\mid\mathcal{X})$. Then, by minimizing the KL divergence, we can approximate the target posterior distribution with a closely matched variational network. Since the left-hand side of Equation (A.2) is irrelevant to the variational network $q(\mathcal{Z}\mid\mathcal{X})$, minimizing the KL divergence is equivalent to maximizing the evidence lower bound (ELBO) given by:

$$\begin{aligned}\mathrm{ELBO}\,(q(\mathcal{Z}\mid\mathcal{X})) &= \mathbb{E}_q\left[\log p(\mathcal{X},\mathcal{Z})\right] - \mathbb{E}_q\left[\log q(\mathcal{Z}\mid\mathcal{X})\right] \\ &= \mathbb{E}_q\left[\log p(\mathcal{Z}\mid\mathcal{X})\right] + \mathbb{E}_q\left[\log p(\mathcal{X})\right] - \mathbb{E}_q\left[\log q(\mathcal{Z}\mid\mathcal{X})\right].\end{aligned} \tag{A.3}$$

Substituting Equation (A.1) into the above equation yields the analytical expression for the ELBO. By maximizing this ELBO, the latent parameters of the model and the variational network can be estimated concurrently using an optimization algorithm, such as Adam (Kingma and Ba 2014).

First, let us describe the application of deep VI to estimate the parameters of a sentiment-oriented topic model. For clarity, we take positive topics as an example here. Consider an item a_j; the set of its positive FOPs is denoted by the vector $\boldsymbol{w}_j^+$. In our model, neural networks function as variational networks that input these FOPs and output a variational distribution representing the positive topics associated with a_j. This process unfolds as follows:

$$\begin{aligned}\boldsymbol{\mu}_{\gamma^+}(\boldsymbol{w}_j^+) &= f_\mu(\boldsymbol{w}_j^+),\\ \boldsymbol{\sigma}^2_{\gamma^+}(\boldsymbol{w}_j^+) &= \exp(f_\sigma(\boldsymbol{w}_j^+)),\end{aligned} \tag{A.4}$$

where $f_\mu(\cdot)$ and $f_\sigma(\cdot)$ are fully connected feedforward neural networks that include batch normalization and dropout techniques to mitigate overfitting. The function $\exp(\cdot)$ ensures the variance values remain positive. Subsequently, the positive topic representation $\boldsymbol{\gamma}_j^+$ is sampled from the diagonal Gaussian distribution in the following manner:

$$\begin{aligned}\boldsymbol{\gamma'}_j^+ &\sim \mathcal{N}(\boldsymbol{\mu}_{\gamma^+}(\boldsymbol{w}_j^+), \boldsymbol{\sigma}^2_{\gamma^+}(\boldsymbol{w}_j^+)),\\ \boldsymbol{\gamma}_j^+ &= softmax(\boldsymbol{\gamma'}_j^+).\end{aligned} \tag{A.5}$$

However, direct sampling methods associate uncertainties with parameters that require optimization, which poses challenges for model inference. To address this issue, the reparameterization strategy is employed, effectively decoupling the sampling-induced uncertainty from the model parameters (Srivastava and Sutton 2016). Typically, an auxiliary variable $\boldsymbol{\epsilon} \in \mathbb{R}^{\mathcal{K}}$, which follows a multivariate standard Gaussian distribution, is introduced to encapsulate all uncertainties related to the sampling process. Subsequently, the latent representation $\boldsymbol{\gamma}_j^+$ is sampled using the following method:

$$\begin{aligned}\boldsymbol{\epsilon} &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}),\\ \boldsymbol{\gamma'}_j^+ &\sim \boldsymbol{\mu}_{\gamma^+}(\boldsymbol{w}_j^+) + \boldsymbol{\sigma}_{\gamma^+}(\boldsymbol{w}_j^+) \odot \boldsymbol{\epsilon},\\ \boldsymbol{\gamma}_j^+ &= softmax(\boldsymbol{\gamma'}_j^+).\end{aligned} \tag{A.6}$$

where $\mathbf{I}$ represents the identity matrix, and $\boldsymbol{\sigma}_{\gamma^+}(\boldsymbol{w}_j^+)$ denotes the standard deviation vector. Consequently, parameters associated with the variational networks $f_\mu(\cdot)$ and $f_\sigma(\cdot)$ can be efficiently optimized using gradient-based methods. Similarly, the negative topic representation $\boldsymbol{\gamma}_j^-$ for item a_j can be sampled using the same process.

Let $q_{\Theta^+}(\boldsymbol{\gamma}_j^+ \mid \boldsymbol{w}_j^+)$ and $q_{\Theta^-}(\boldsymbol{\gamma}_j^- \mid \boldsymbol{w}_j^-)$ represent the variational networks for the topic representations, with Θ^+ and Θ^- as the respective parameter sets in these networks. For each item a_j, the Evidence Lower Bound (ELBO) is expressed as follows:

$$\begin{aligned}ELBO =& \mathbb{E}_{q_{\Theta^+}}[\log p(\boldsymbol{w}_j^+, \boldsymbol{\gamma}_j^+ \mid \{\boldsymbol{\phi}_k^+\})] + \mathbb{E}_{q_{\Theta^-}}[\log p(\boldsymbol{w}_j^-, \boldsymbol{\gamma}_j^- \mid \{\boldsymbol{\phi}_k^-\})] - \mathbb{E}_{q_{\Theta^+}}[\log q_{\Theta^+}(\boldsymbol{\gamma}_j^+ \mid \boldsymbol{w}_j^+)] - \mathbb{E}_{q_{\Theta^-}}[\log q_{\Theta^-}(\boldsymbol{\gamma}_j^- \mid \boldsymbol{w}_j^-)]\\ =& \mathbb{E}_{q_{\Theta^+}}[\log p(\boldsymbol{w}_j^+ \mid \boldsymbol{\gamma}_j^+, \{\boldsymbol{\phi}_k^+\})] + \mathbb{E}_{q_{\Theta^-}}[\log p(\boldsymbol{w}_j^- \mid \boldsymbol{\gamma}_j^-, \{\boldsymbol{\phi}_k^-\})]\\ &- KL(q_{\Theta^+}(\boldsymbol{\gamma}_j^+ \mid \boldsymbol{w}_j^+) || p(\boldsymbol{\gamma}_j^+)) - KL(q_{\Theta^-}(\boldsymbol{\gamma}_j^- \mid \boldsymbol{w}_j^-) || p(\boldsymbol{\gamma}_j^-)),\end{aligned} \tag{A.7}$$

where $KL(\cdot||\cdot)$ denotes the KL divergence between two distributions. It should be noted that since the variational networks $q_{\Theta^+}(\boldsymbol{\gamma}_j^+ \mid \boldsymbol{w}_j^+)$ and $q_{\Theta^-}(\boldsymbol{\gamma}_j^- \mid \boldsymbol{w}_j^-)$ are implemented using neural networks, the expectation components of the ELBO lack analytical expressions. In such instances, the Monte Carlo method can estimate these expectations by sampling $\boldsymbol{\gamma}_j^+$ and $\boldsymbol{\gamma}_j^-$ from the variational networks. Additionally, employing the reparameterization technique as in Equation (B.4) significantly reduces variance within these networks, enabling the estimation of expectation values with just a single sample (Kingma and Welling 2014). Consequently, with a single sample of the positive representation $\dot{\boldsymbol{\gamma}}_j^+$ and the negative representation $\dot{\boldsymbol{\gamma}}_j^-$, the ELBO can be reformulated as follows:

$$ELBO = \log p(\boldsymbol{w}_j^+ \mid \dot{\boldsymbol{\gamma}}_j^+, \{\boldsymbol{\phi}_k^+\}) + \log p(\boldsymbol{w}_j^- \mid \dot{\boldsymbol{\gamma}}_j^-, \{\boldsymbol{\phi}_k^-\}) - KL(q_{\Theta^+}(\boldsymbol{\gamma}_j^+ \mid \boldsymbol{w}_j^+)||p(\boldsymbol{\gamma}_j^+)) - KL(q_{\Theta^-}(\boldsymbol{\gamma}_j^- \mid \boldsymbol{w}_j^-)||p(\boldsymbol{\gamma}_j^-)). \tag{A.8}$$

Then, parameters in the sentiment-oriented topic model, such as Θ^+, Θ^-, $\{\boldsymbol{\phi}_k^+\}$, and $\{\boldsymbol{\phi}_k^-\}$, can be estimated by maximizing the ELBO value using gradient-based optimization methods.

We now focus to inferring parameters associated with users' rating behaviors. Similarly, the mixture of motivations $\boldsymbol{\theta}_{i,t}$ for user d_i during time period t can be sampled from the appropriate variational networks. The variational network $q_\Phi(\boldsymbol{\theta}_{i,t} \mid \boldsymbol{b}_{i,t})$, which pertains to users' rating behaviors, is defined by the parameter set Φ. For each user d_i, the ELBO related to her behaviors can be derived as follows:

$$\begin{aligned} ELBO =& \mathbb{E}_{q_\Phi}[\log p(\{\boldsymbol{\theta}_{i,t}\}, \{x_{i,t,n}\}, \{r_{i,t,n}\} \mid \{\boldsymbol{\beta}_m\}, \{\mathbf{u}_{i,m}\})] - \mathbb{E}_{q_\Phi}[\log q_\Phi(\{\boldsymbol{\theta}_{i,t}\} \mid \{\boldsymbol{b}_{i,t}\})] \\ =& \mathbb{E}_{q_\Phi}[\log p(\{r_{i,t,n}\} \mid \{\boldsymbol{\theta}_{i,t}\}, \{x_{i,t,n}\}, \{\boldsymbol{\beta}_m\}, \{\mathbf{u}_{i,m}\})] + \mathbb{E}_{q_\Phi}[\log p(\{x_{i,t,n}\} \mid \{\boldsymbol{\theta}_{i,t}\}, \{\boldsymbol{\beta}_m\})] \\ & - KL(\log q_\Phi(\{\boldsymbol{\theta}_{i,t}\} \mid \{\boldsymbol{b}_{i,t}\})||p(\{\boldsymbol{\theta}_{i,t}\})) \end{aligned} \tag{A.9}$$

Here too, the challenge stems from the intractable nature of the variational network $q_\Phi(\boldsymbol{\theta}_{i,t} \mid \boldsymbol{b}_{i,t})$, which results in the expectation terms of the ELBO lacking explicit expressions. Consequently, we continue to utilize the Monte Carlo method, estimating these expectations by sampling from the variational network to acquire samples of the motivational distributions $\{\boldsymbol{\theta}_{i,t}\}$ for user d_i. A sample of the user's motivational proportions across all time periods is denoted by $\{\dot{\boldsymbol{\theta}}_{i,t}\}$, and the corresponding ELBO is reformulated as follows:

$$\begin{aligned} ELBO =& \log p(\{r_{i,t,n}\} \mid \{\dot{\boldsymbol{\theta}}_{i,t}\}, \{x_{i,t,n}\}, \{\boldsymbol{\beta}_m\}, \{\mathbf{u}_{i,m}\}) + \log p(\{x_{i,t,n}\} \mid \{\dot{\boldsymbol{\theta}}_{i,t}\}, \{\boldsymbol{\beta}_m\}) - KL(\log q_\Phi(\{\boldsymbol{\theta}_{i,t}\} \mid \{\boldsymbol{b}_{i,t}\})||p(\{\boldsymbol{\theta}_{i,t}\})) \\ =& \sum_{t=1}^{\mathcal{T}} \sum_{n=1}^{\mathcal{N}_{i,t}} \log p(r_{i,t,n} \mid x_{i,t,n}, \dot{\boldsymbol{\theta}}_{i,t}, \{\boldsymbol{\beta}_m\}, \{\mathbf{u}_{i,m}\}) + \sum_{t=1}^{\mathcal{T}} \sum_{n=1}^{\mathcal{N}_{i,t}} \log p(x_{i,t,n} \mid \dot{\boldsymbol{\theta}}_{i,t}, \{\boldsymbol{\beta}_m\}) \\ & - \sum_{t=1}^{\mathcal{T}} KL(\log q_\Phi(\boldsymbol{\theta}_{i,t} \mid \boldsymbol{b}_{i,t})||p(\boldsymbol{\theta}_{i,t})). \end{aligned} \tag{A.10}$$

To prevent overfitting on the training dataset, we incorporate regularization terms into the user's preference models $\{\mathbf{u}_{i,m}\}$ within the specified ELBO to limit its complexity. The final optimization objective is expressed as follows:

$$\begin{aligned} ELBO_{\text{norm}} =& \sum_{t=1}^{\mathcal{T}} \sum_{n=1}^{\mathcal{N}_{i,t}} \log p(r_{i,t,n} \mid x_{i,t,n}, \dot{\boldsymbol{\theta}}_{i,t}, \{\boldsymbol{\beta}_m\}, \{\mathbf{u}_{i,m}\}) + \sum_{t=1}^{\mathcal{T}} \sum_{n=1}^{\mathcal{N}_{i,t}} \log p(x_{i,t,n} \mid \dot{\boldsymbol{\theta}}_{i,t}, \{\boldsymbol{\beta}_m\}) \\ & - \sum_{t=1}^{\mathcal{T}} KL(\log q_\Phi(\boldsymbol{\theta}_{i,t} \mid \boldsymbol{b}_{i,t})||p(\boldsymbol{\theta}_{i,t})) + \alpha \sum_{m=1}^{\mathcal{M}} ||\mathbf{u}_{i,m}||, \end{aligned} \tag{A.11}$$

where α is treated as a hyper-parameter optimized through cross-validation. Subsequently, parameters related to users' ratings, including Φ, $\{\boldsymbol{\beta}_m\}$, and $\{\mathbf{u}_{i,m}\}$, are estimated by maximizing the objective using gradient-based optimization methods. Within the deep variational inference framework, the prior distributions of the item-topic parameters $\{\boldsymbol{\gamma}_j^+\}$, $\{\boldsymbol{\gamma}_j^-\}$, and the user-motivation parameters $\{\boldsymbol{\theta}_{i,t}\}$, denoted by $p(\boldsymbol{\gamma}_j^+)$, $p(\boldsymbol{\gamma}_j^-)$, and $p(\boldsymbol{\theta}_{i,t})$, are assumed to follow standard Gaussian distributions. Additionally, a softmax layer is incorporated to ensure the probabilistic properties of these distributions. This configuration enables the inference of the ELBO objective function to obtain an analytical expression (Kingma and Welling 2014, Yang et al. 2023).

From an operational perspective, to manage the variability in the lengths of user behavior vectors $\boldsymbol{b}_{i,t}$ during parameter inference, we employ padding and masking techniques. Specifically, all vectors are padded with zeros to match the length of the longest vector in the dataset. This standardization facilitates parallel processing of data from the same batch, significantly accelerating the optimization process. Additionally, each vector is masked to represent the actual length of data before padding. This mask is used to construct the objective, ensuring that parameter inference is conducted only on the non-padded elements of the vectors. By enabling efficient batch processing, these techniques enhance computational efficiency and ensure that our model can handle large-scale datasets swiftly and effectively.

A.2. Predictive Recommendation

The estimated values of latent parameters provide the basis for deriving interpretable insights into empirical user behaviors and predicting their preferences for unrated items. We use $\hat{\zeta}$ to represent the estimated value of ζ, where ζ is one of the latent parameters in our model. Specifically, the top FOP triplets associated with parameters $\{\hat{\boldsymbol{\phi}}_k^+\}$ and $\{\hat{\boldsymbol{\phi}}_k^-\}$ capture semantic information regarding various aspects of the items associated with positive and negative topics. Likewise, parameters $\{\hat{\boldsymbol{\beta}}_m\}$ help understand users' specific motivations by summarizing frequently occurring items. In contrast, the parameters related to item representations and user contextual preference models, including $\{\hat{\boldsymbol{\gamma}}_j^+\}$, $\{\hat{\boldsymbol{\gamma}}_j^-\}$, $\{\hat{\boldsymbol{\beta}}_m\}$, $\{\hat{\boldsymbol{\theta}}_{i,t}\}$, and $\{\hat{\mathbf{u}}_{i,m}\}$, enable predictive recommendations for items with disclosed preferences.

We now describe the procedure for generating predictive recommendations for users' unrated items based on the inferred parameters. For each unrated item $a_j \in \mathcal{A} \setminus \mathcal{A}_{i,t}^R$, let $z(a_j)$ represent the underlying motivation that drives a user to select item a_j. We employ Bayes' theorem to infer the posterior probability of user d_i choosing item a_j activated by motivation m (i.e., $z(a_j) = m$) in time period t as follows:

$$p(z(a_j) = m \mid a_j, \hat{\boldsymbol{\theta}}_{i,t}, \hat{\boldsymbol{\beta}}_m) = \frac{p(a_j \mid z(a_j) = m, \hat{\boldsymbol{\beta}}_m) p(z(a_j) = m \mid \hat{\boldsymbol{\theta}}_{i,t})}{\sum_{l=1}^{\mathcal{M}} \mathrm{P}(a_j \mid z(a_j) = l, \hat{\boldsymbol{\beta}}_l) p(z(a_j) = l \mid \hat{\boldsymbol{\theta}}_{i,t})} = \frac{\hat{\beta}_{m,j} \cdot \hat{\theta}_{i,t,m}}{\sum_{l=1}^{\mathcal{M}} \hat{\beta}_{l,j} \cdot \hat{\theta}_{i,t,l}}. \tag{A.12}$$

Let $\hat{\boldsymbol{\tau}}_j = \{\hat{\tau}_{j,k}\}_{k=1}^{\mathcal{K}^+ + \mathcal{K}^-}$ denote the posterior mean of the characteristic vector of item a_j. Then, we compute the user's predictive preference for item $a_j \in \mathcal{A} \setminus \mathcal{A}_{i,t}^R$ by calculating the expected comprehensive value for a_j based on the posterior distribution of motivations as follows:

$$\hat{\mathcal{U}}(a_j) = \sum_{m=1}^{\mathcal{M}} p(z(a_j) = m \mid a_j, \hat{\boldsymbol{\theta}}_{i,t}, \hat{\boldsymbol{\beta}}_m) \cdot \hat{\mathbf{u}}_{i,m}^{\mathrm{T}} \hat{\boldsymbol{\tau}}_j = \sum_{m=1}^{\mathcal{M}} \frac{\hat{\beta}_{m,j} \cdot \hat{\theta}_{i,t,m}}{\sum_{l=1}^{\mathcal{M}} \hat{\beta}_{l,j} \cdot \hat{\theta}_{i,t,l}} \cdot \hat{\mathbf{u}}_{i,m}^{\mathrm{T}} \hat{\boldsymbol{\tau}}_j. \tag{A.13}$$

We emphasize the importance of utilizing the expected comprehensive value $\hat{\mathcal{U}}(a_j)$ as the user's predicted preference rather than constructing it based solely on a single rating. This method enables us to factor in the purchasing probabilities corresponding to items activated by different motivations. This is an essential consideration for businesses seeking to optimize their revenue. Traditional recommender systems often prioritize recommending higher-rated items to users, but they fail to account for the likely relevance of these items to the users. Consider a scenario where a user rates an expensive smartphone highly as a prospective gift to herself. However, if empirical data suggest a low likelihood of her purchasing or repurchasing this product, recommending it to her would be inappropriate despite its high rating. Our approach, grounded in the expected comprehensive value, solves this problem by considering both the user's purchasing probabilities and their affinity towards an item. It thus offers a more holistic and accurate strategy for constructing personalized recommendation lists for users.

Appendix B. Experimental Results on Google Maps and MovieLens Datasets

Experimental results on the Google Maps and MovieLens datasets are presented in Section 4 (Empirical Experiment) of the SSRN version, available at `https://ssrn.com/abstract=4606645`.

Appendix C. Benchmark Models

To assess the predictive performance of our model in user ratings, we utilize the following benchmark models:

- *Model 1 (without dynamic effects)*. This model assumes that the motivations underlying all rating behaviors of the user d_i are generated by the same motivation proportion vector $\boldsymbol{\theta}_i$, thus disregarding the dynamic effects in the user's preferences. The other components of the original model remain unchanged.

- *Model 2 (without dynamic and context-dependent effects)*. This model posits that all rating behaviors of user d_i follow a single preference model $\mathbf{u}_i$. It omits the variations in user preferences towards distinct item aspects under different motivations and the temporal fluctuations over consecutive time periods. The other components of the original model remain unchanged.

- *Probabilistic Matrix Factor* (PMF) (Mnih and Salakhutdinov 2007). As a prevalent statistical machine learning method, this model is predominantly used for tackling challenges within recommender systems. It decomposes the user-item rating matrix into two low-rank matrices, each capturing user-specific and item-specific features. The multiplication of these two matrices approximates the original rating matrix.

- *UTADIS* (Doumpos and Zopounidis 2002). This method is widely used in Multiple Criteria Decision Analysis (MCDA). It quantifies the performance of different items across various criteria and determines item ratings through an additive value function and a set of thresholds. When inferring the model's parameters, it characterizes the discrepancy between the predicted rating and the user's actual rating by introducing bias decision variables. The optimal model parameters are then obtained by minimizing these biases.

- *Naive Sentiment Analysis* (NSA). User reviews frequently convey sentiment polarity, reflecting satisfaction or dissatisfaction with an item. This method leverages the sentiment polarity expressed in reviews to directly predict product ratings. However, the effectiveness of this approach depends on the availability of textual analysis of user reviews; thus, it is unsuitable for prediction tasks where corresponding reviews are unavailable.

As our model is designed to predict user ratings and construct representations for items and users, evaluating it solely through a train-predict pattern for predicting user ratings is inadequate. Inspired by Büschken and Allenby (2016), this paper incorporates two variants of topic models to compare their performance with our model in building representations of items and users in an unsupervised learning manner. These two unsupervised benchmark models are as follows:

- *Probabilistic Latent Semantic Analysis* (pLSA) (Hofmann 1999). The pLSA (also known as Probabilistic Latent Semantic Indexing) utilizes matrix factorization to statistically analyze relationships between documents and their constituent terms by probabilistically modeling latent topics. This method yields interpretable results through a probabilistic framework that elucidates the association of words with underlying topics in documents.

- *Dirichlet Multinomial Mixture* (DMM) (Yin and Wang 2014). The DMM model, a variant of LDA, is specifically designed for analyzing short texts. It posits that all words in a document stem from a single topic. Consequently, the sparse co-occurrences of words within each short document significantly enhance the quality of the derived latent topics by contributing uniformly to the same topic.

Appendix D. Robustness Analysis

In this section, we evaluate the robustness of our model across various datasets, examining its performance in relation to different levels of user behavior sparsity under various time period settings and selection bias. This analysis allows us to comprehensively understand the model's performance across diverse scenarios, validating its applicability and stability, and ensuring its effectiveness and reliability in practical applications. To assess our model's robustness against data sparsity, we categorize users into three groups: Dense, Moderate, and Sparse, based on their average rating behavior per time period within the estimation set. We subsequently compare the out-of-sample rating prediction performance of our model with benchmark models across these groups to evaluate the impact of data sparsity on various methods. Note that the NSA model's performance is independent of user data sparsity as it does not require training on the estimation set. Consequently, we exclude it from this analysis. Table 1 displays the RMSE values for our model and various benchmark models across out-of-sample sets for users at different levels of sparsity. The results are two-fold. On the one hand, as user data sparsity increases, the performance of all models declines, consistent with the expectation that less training data introduces greater uncertainty in learning user preferences, thus impairing the accuracy of user rating predictions. On the other hand, across nearly all sparsity levels, our model consistently outperforms the benchmark models, achieving low RMSE values across all datasets. These results underscore the robustness and superior performance of our model.

Table 1 RMSE Values for Our Model and Benchmark Models across Users with Different Sparsity Levels.

Method	Amazon			Google Maps			MovieLens		
	Dense	Moderate	Sparse	Dense	Moderate	Sparse	Dense	Moderate	Sparse
Complete Model	1.065	1.158	1.306	0.786	0.833	0.856	0.877	0.889	0.894
Model 1	1.070	1.159	1.308	0.787	0.841	0.867	0.876	0.897	0.903
Model 2	1.079	1.161	1.309	0.790	0.839	0.870	0.895	0.914	0.923
PMF	1.125	1.183	1.284	0.818	0.873	0.912	0.920	0.940	0.948
UTADIS	1.179	1.290	1.491	0.839	0.910	0.949	0.926	0.947	0.958

Additionally, we assess our model's performance across various datasets for different granularities of time periods, an important hyper-parameter in the analysis of dynamic user preferences. Typically, we segment our datasets based on time period lengths of half a year, one year, and two years, and apply our complete model to analyze its performance on out-of-sample sets for these respective granularities. Table 2 presents the MAE and RMSE values achieved by our model across all out-of-sample sets for different time period lengths. It can be observed that the performance of our model does not significantly vary with different granularities of time periods. In all cases, it achieves better predictive accuracy compared to other benchmark models

Table 2 MAE and RMSE Values for Our Model across Different Time Period Lengths.

Length of Time Period	Amazon		Google Maps		MovieLens	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Half a year	0.849	1.154	0.602	0.832	0.677	0.886
One year	0.847	1.150	0.595	0.831	0.678	0.888
Two years	0.845	1.153	0.590	0.830	0.673	0.883

(see Table 9 in the main text). This observation indicates that our model is insensitive to the granularity of time periods, thereby increasing its potential robustness in real-world applications.

Our dataset may exhibit selection bias, as the decision to leave a review after purchase is not entirely random. To assess whether selection bias influences the model's performance, we conduct a robustness check using the Heckman correction (Heckman 1979) to simulate the "selection" stage of user review-leaving behavior. Since our dataset lacks prior information about a user's consideration set for providing reviews, we augment it by randomly selecting, for each item a user has reviewed, an additional item that the user has not reviewed. We then estimate a logistic regression model based on user characteristics (e.g., number of reviews provided, average review length, and average rating) and item information (e.g., number of reviews received, average rating, item category, and price). Using the estimated model, we calculate the probability that each review appears in the dataset. To address potential bias, we apply inverse probability weighting to each review based on its estimated probability of being observed and sample accordingly to construct a review corpus. This adjusted corpus is used to generate item characteristic vectors, which serve as inputs to our user rating model for performance comparison with the original results. The findings in Table 3 reveal that selection correction has no significant impact on model performance, as indicated by similar rating metrics. Thus, we present the original model results in the performance comparison section without selection correction, as no additional adjustments are necessary for the dataset.

Table 3 MAE and RMSE Values of Our Model With and Without Selection Correction.

Setting	Amazon		Google Maps		MovieLens	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Original results	0.840	1.157	0.593	0.825	0.673	0.883
Results with selection correction	0.842	1.158	0.598	0.828	0.669	0.880

The insignificant impact of the selection correction could be attributed to several factors. For instance, the review corpus may already provide sufficient information to construct characteristic vectors that capture item performance, or the current dataset might lack the capacity to support training a logistic regression model that accurately captures users' review-leaving behaviors. Various external factors, such as social network influence and merchant promotions, can affect a user's decision to leave an item review. Due to the absence of data in our dataset that characterizes these external factors, developing a mechanism to fully explain users' selective reviewing behavior is nearly impossible.

References

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022.

Büschken, J. and Allenby, G. M. (2016). Sentence-based text analysis for customer reviews. *Marketing Science*, 35(6):953–975.

Doumpos, M. and Zopounidis, C. (2002). *Multicriteria Decision Aid Classification Methods*. Springer Science & Business Media, New York, USA.

Heckman, J. (1979). Sample selection bias as a specification error. *Econometrica*.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57.

Kingma, D. P. and Ba, J. (2014). Adam: a method for stochastic optimization 3rd int. In *Conf. for Learning Representations, San*.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. *Stat*, 1050:1.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Mnih, A. and Salakhutdinov, R. R. (2007). Probabilistic matrix factorization. *Advances in Neural Information Processing Systems*, 20.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT Press.

Srivastava, A. and Sutton, C. (2016). Autoencoding variational inference for topic models. In *International Conference on Learning Representations*.

Yang, Y., Zhang, K., and Fan, Y. (2023). sdtm: A supervised bayesian deep topic model for text analytics. *Information Systems Research*, 34(1):137–156.

Yin, J. and Wang, J. (2014). A Dirichlet multinomial mixture model-based approach for short text clustering. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 233–242.