8-2018

Probabilistic collaborative representation learning for personalized item recommendation

Aghiles SALAH
Singapore Management University, asalah@smu.edu.sg

Hady W. LAUW
Singapore Management University, hadywlauw@smu.edu.sg

Probabilistic Collaborative Representation Learning for Personalized Item Recommendation

Aghiles Salah and **Hady W. Lauw**
School of Information Systems
Singapore Management University, Singapore
{asalah, hadywlauw}@smu.edu.sg

Abstract

We present Probabilistic Collaborative Representation Learning (PCRL), a new generative model of user preferences and item contexts. The latter builds on the assumption that relationships among items within contexts (e.g., browsing session, shopping cart, etc.) may underlie various aspects that guide the choices people make. Intuitively, PCRL seeks representations of items reflecting various regularities between them that might be useful at explaining user preferences. Formally, it relies on Bayesian Poisson Factorization to model user-item interactions, and uses a multilayered latent variable architecture to learn representations of items from their contexts. PCRL seamlessly integrates both tasks within a joint framework. However, inference and learning under the proposed model are challenging due to several sources of intractability. Relying on the recent advances in approximate inference/learning, we derive an efficient variational algorithm to estimate our model from observations. We further conduct experiments on several real-world datasets to showcase the benefits of the proposed model.

1 INTRODUCTION

With pervasive digitization of marketplaces and services, we now make most of our consumption choices online. Relieved from the inventory limitation of a physical storefront, online providers are able to offer a mind-boggling array of choices numbering in the thousands to millions. To help users in navigating this sea of choices, modern applications rely heavily on recommender systems to deliver a personalized ranking or selection of items to each user according to her preferences.

There are various approaches to recommender systems, including memory-based and model-based approaches (Sarwar et al., 2001). At the heart of the more prevalent model-based approach is learning a latent representation for every user and every item. Such a latent representation places a user or an item in the "feature" space of preferences, such that when two related items share similar representations or "features", a user who prefers one likely also prefers the other. Further recommendation predictions are based on these latent representations.

Much of the previous work seek to learn these representations from historical behavioral data, such as ratings, clicks, purchases, etc. (usually organized into a user-item interaction or preference matrix). For instance, the widespread Matrix Factorization (MF) (Mnih and Salakhutdinov, 2008; Hu et al., 2008; Koren et al., 2009) derives user and item latent representations in the form of low dimensional vectors by decomposing the preference matrix. The bilinear combination of user and item's latent factors can be used to predict unknown preferences.

The limitation of learning these representations from historical behaviors is the sparsity of such data. The long-tail effect (Park and Tuzhilin, 2008) means that most items have been adopted by few users. Moreover, given the rapid expansion of catalogues, there are continually new items with scant record of historical consumption. One consequence of this sparsity is that closely related items may not be mapped to the same direction in the latent space, as they might not have been rated by the same users. As such, historical consumption data alone may not suffice for learning effective item representations.

In some real-world scenarios, there may be known some auxiliary information on how items are likely related to one another. For example, a user interested in a particular shirt may also be interested in a matching pair of jeans. Moreover, such relatedness among items may not have to be explicitly stated, and could be implicitly inferred from such indicative events as whether items are placed

within the same shopping carts, are browsed within the same session, etc. Such item-item relationships constitute valuable information that would otherwise not easily be derivable from similarities in product attributes alone. We thus seek to enrich the learned item representation to also incorporate such item-item relationships, to supplement the sparse user-item interactions.

Representation learning (Bengio et al., 2013) is of interest to learn features or representations from different data, such as images, text, etc. Recent techniques rely on deep neural networks to learn compositional representations. While inspired by this promising approach, our work is set apart in that we are interested not only on extracting objective features of items, but more importantly also those that could help describe user preferences effectively. Therefore, instead of relying on representation learning solely or separately, given the efficacy of probabilistic models for collaborative filtering, we propose to conjoin the representation learning from item-item contextual relationships, and collaborative filtering from user-item interactions, within a unified model.

In this paper, we develop Probabilistic Collaborative Representation Learning (PCRL), which seeks to learn item representations both *contextually* based on their relatedness with other items, as well as *collaboratively* based on their interactions/adoptions by users. For the former, PCRL uses a multilayered (hierarchical) latent variable structure, with a Poisson likelihood and Gamma distributed layers, to model the item's context (e.g., shopping cart, session). For the latter, PCRL relies on Poisson Factorization (PF) for decomposing users' interactions with items. As shown in (Gopalan et al., 2015), PF realistically models user preferences, fits well to sparse data thanks to the Poisson's mathematical form, and it substantially outperforms previous state-of-the-art Gaussian likelihoods-based MF models (Mnih and Salakhutdinov, 2008; Shan and Banerjee, 2010; Koren et al., 2009) for item recommendation.

PCRL joins both sources of data through a shared item latent space within a probabilistic generative model. Intuitively, the collaborative PF component can guide the contextual representation learning process to focus on extracting features that are relevant for predicting the preference information. The contextual representation learning component in turn will encourage the PF part to rely on items' contexts to explain user preferences, which would supplement the lack of user-item interactions.

Exact inference under the PCRL model is very challenging due to various sources of intractability. To overcome this difficulty we rely on recent innovations in approximate inference/learning and derive an efficient variational algorithm to estimate PCRL from observed user preferences and item contexts. Empirical results on several real-world datasets reflect the benefits of PCRL in terms of both personalized recommendation and item representation learning.

2 RELATED WORK

The sparsity of preference data has driven many to extend Matrix Factorization (MF) models (Mnih and Salakhutdinov, 2008; Hu et al., 2008; Koren et al., 2009) beyond user-item interactions, and leverage auxiliary information, such as social networks (Ma et al., 2008; Zhou et al., 2012; Rao et al., 2015), product taxonomy (Koenigstein et al., 2011), item content (Wang and Blei, 2011), etc. However, these are mostly still within the framework of MF. For instance, Collective Matrix Factorization (Singh and Gordon, 2008), which co-factorized multiple data matrices, is a popular approach in the recommendation literature to jointly model several sources of information.

Yet other approaches, similarly to ours, use graphical models to join different modalities. Wang and Blei (2011) developed Collaborative Topic Regression (CTR), which composes a topic model, Latent Dirichlet Allocation (LDA), with probabilistic matrix factorization to model texts (articles) and user (reader) preferences. Along the same line, Wang et al. (2015); Li and She (2017) proposed alternatives to CTR where probabilistic auto-encoder, is substituted for LDA for modeling text.

We focus on incorporating item relatedness, a modality mostly neglected by previous personalized recommendation models. Notable exceptions include CoFactor (Liang et al., 2016) and Matrix Co-Factorization (MCF) (Park et al., 2017), which used the principle of collective MF based on Gaussian likelihoods. In contrast, we build on Bayesian Poisson Factorization (PF), and we further investigate another architecture for leveraging the item's contexts with new modeling perspectives. In experiments, we compare to the more recent MCF that learns from item network as a baseline. CoFactor learns not from an external auxiliary source, but rather from item-item relations induced from the user-item interactions.

Since (Gopalan et al., 2015), there is a growing body of work on applying PF (Canny, 2004; Cemgil, 2009) to recommender systems. Gopalan et al. (2014a) developed non-parametric PF. Chaney et al. (2015) incorporated social interactions. Charlin et al. (2015) accounted for user and item evolution over time. Notably, Gopalan et al. (2014b) proposed Collaborative Topic Poisson Factorization (CTPF) to model both article contents and reader preferences. In contrast to CTPF that uses PF to model both the user preferences and auxiliary item information (text), we adopt PF for the user-item interactions only,

and we use a multilayered latent variable structure to learn item representations from auxiliary data (item contexts). The benefits of our modeling architecture would be reflected in the experiments with CTPF as a baseline.

3 PROBABILISTIC COLLABORATIVE REPRESENTATION LEARNING

The observed data that we would learn from are user preferences and item contexts respectively. The former are organized into a user-item preference matrix of size $U \times I$, denoted $\mathbf{X} = (x_{ui})$, where x_{ui} is the integer rating[1] that user u gave to item i, or zero if no preference was expressed. The contextual interactions among items are encoded in an item-context matrix $\mathbf{C} = (c_{ij})$, of size $I \times J$, where $c_{ij} = 1$ if item j belongs to the context[2] of i, and $c_{ij} = 0$ otherwise. The i^{th} row of this matrix is represented by a vector $\mathbf{c}_i = (c_{i1}, \ldots, c_{iJ})^\top$, where $\top$ denotes the transpose. We will refer to the set of items j such that $c_{ij} > 0$ as the context of item i.

We now describe *Probabilistic Collaborative Representation Learning* or PCRL, a new probabilistic latent variable model for jointly modeling user preferences and item contexts. The intuition is to learn item representations reflecting various contextual relationships between them that are useful for explaining user preferences. Figure 1 depicts PCRL in plate notation.

Contextual Representation Learning. To model representations due to the item contexts (refer to the left portion of Figure 1) we use a multilayer structure similar to Deep Exponential Families (Ranganath et al., 2015). More precisely, PCRL assumes L layers of hidden variables per item: $\mathcal{Z}_i = \{\mathbf{z}_{i,1}, \ldots, \mathbf{z}_{i,L}\}$, such that $\mathbf{z}_{i,\ell} \in \mathbb{R}_+^{K_\ell}$. For a reason that will be clear shortly, we denote $\mathbf{z}_{i,L+1} = \boldsymbol{\beta}_i$. Along with these variables, PCRL has $L + 1$ layers of latent weights shared across items, $\mathcal{W} = \{\mathbf{W}_0, \ldots, \mathbf{W}_L\}$, where $\mathbf{W}_\ell$ is a matrix of size $K_{\ell+1} \times K_\ell$, with $K_0 = J$, and its k^{th} column is denoted by $\mathbf{w}_{\ell,k}$. Effectively, each hidden layer models representations for items based on their contexts. Intuitively, a higher layer encodes a higher level of representational abstraction; $\boldsymbol{\beta}_i$ is the most abstract representation.

The components $z_{i,\ell,k}$ at each hidden layer are Gamma distributed. Note that this choice is not a limitation of our modeling framework. Depending on specific requirements, other types of $z_{i,\ell,k}$ are possible, e.g., Gaussian, and these might differ across layers.

[1]Other user-item interactions indicative of preferences are also possible, e.g., number of clicks.

[2]The definition of "context" is scenario-dependent, e.g., another item j is found in the same shopping cart as item i.

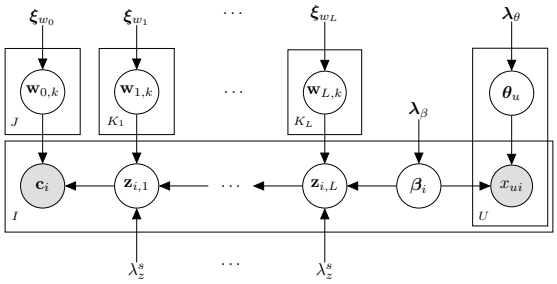

Figure 1: The proposed model PCRL in plate representation, $\boldsymbol{\xi}$ and $\boldsymbol{\lambda} = (\lambda^s, \lambda^r)$ stand for Gaussian and Gamma parameters.

To capture various correlations across layers, including negative ones, we let the weights $\mathbf{W}_\ell$ be real valued with Gaussian priors. These latent variables interact with each other to explain the contextual relationships among items. While several interaction schemes are possible, we mimic neural networks (multilayer perceptron or MLP), and let the mean of the local variable at the current layer to be driven by the current weights and the previous layer as follows:

$$\mathbb{E}(\mathbf{z}_{i,\ell}|\mathbf{W}_\ell, \mathbf{z}_{i,\ell+1}) = a_\ell(\mathbf{z}_{i,\ell+1}^\top \mathbf{W}_\ell) \tag{1}$$

where $a_\ell(x)$ is a function that maps x into the right mean space. Following the nomenclature in the neural network literature, we call it the *activation function*.

Conditional on the lowest layer, $\mathbf{z}_{i,1}$, the components of the item-context vector $\mathbf{c}_i$ are independent Poisson variables, i.e., $\mathbf{c}_i \sim p(\mathbf{c}_i|\mathbf{z}_{i,1}, \mathcal{W}) = \prod_j p(c_{ij}|\mathbf{z}_{i,1}, \mathcal{W})$, and

$$p(c_{ij}|\mathcal{Z}, \mathcal{W}, \boldsymbol{\beta}) = \text{Poisson}(\mathbf{z}_{i,1}^\top \mathbf{w}_{0,j}) \tag{2}$$

where $\mathbf{w}_{0,j}$ denotes the j^{th} column of the matrix $\mathbf{W}_0$.

Collaborative Poisson Factorization. To model user preferences (refer to the right portion of Figure 1), PCRL relies on Poisson factorization, i.e.,

$$x_{ui}|\boldsymbol{\theta}, \boldsymbol{\beta} \sim \text{Poisson}(\boldsymbol{\theta}_u^\top \boldsymbol{\beta}_i), \tag{3}$$

where $\boldsymbol{\theta}_u^\top \in \mathbb{R}_+^K$ and $\boldsymbol{\beta}_i^\top \in \mathbb{R}_+^K$ are latent variables referred to as the vectors of user preferences and item attributes respectively. Similar to the original Bayesian Poisson factorization, we let the user preferences θ_{uk} and item attributes β_{ik} be Gamma random variables—throughout the paper, we use the shape and rate parameterization of the Gamma distribution.

Unified Generative Model. The intuition behind this multilayer architecture and sharing $\boldsymbol{\beta}$ between the collaborative and contextual parts, is to let the latent variables $\mathcal{Z}$ and $\mathcal{W}$, at the intermediate layers, absorb various item-context patterns encoded in $\mathbf{C}$, while encouraging the item latent attributes $\boldsymbol{\beta}$ to capture only those patterns

which are useful for explaining user preferences. The corresponding generative process is as follows:

1. Draw user preferences: $\theta_{uk} \sim \text{Gamma}(\lambda_\theta^s, \lambda_\theta^r)$.
2. For each item i
   (a) Draw its attributes: $\beta_{ik} \sim \text{Gamma}(\lambda_\beta^s, \lambda_\beta^r)$
   (b) For each layer ℓ, for $k \in \{1, \ldots, K_\ell\}$:
      i. Draw $\mathbf{w}_{\ell,k} \sim \mathcal{N}(\boldsymbol{\mu}_\ell, \boldsymbol{\sigma}_\ell^2 \mathbf{I}_{K_{\ell+1}})$
      ii. Draw $z_{i,\ell,k} \sim \text{Gamma}(\lambda_z^s, \frac{\lambda_z^s}{a_\ell(\mathbf{z}_{\ell+1}^\top \mathbf{w}_{\ell,k})})$
   (c) For $j \in \{1, \ldots, J\}$,
      i. Draw $\mathbf{w}_{0,j} \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0^2 \mathbf{I}_{K_1})$
      ii. Draw $c_{ij} \sim \text{Poisson}(a_0(\mathbf{z}_{i,1}^\top \mathbf{w}_{0,j}))$
3. For each user-item pair (u,i) sample a preference: $x_{ui} \sim \text{Poisson}(\boldsymbol{\theta}_u^\top \boldsymbol{\beta}_i)$,

where $\mathbf{I}_K$ stands for the identity matrix of size K. In practice, we use the standard multivariate isotropic Gaussian as the prior over each variable $\mathbf{w}_{\ell,k}$. Further, for efficiency purposes, we will make the latent variables $\mathbf{z}_{i,\ell}$ for $\ell \in \{1, \ldots, L\}$ deterministic by taking λ_z^s to infinity.

In principle PCRL should place high probability on item factors $\boldsymbol{\beta}$ reflecting various item relationship patterns that are useful at explaining user preferences.

Connections to Existing Models. In unifying item contexts and user-item preferences, PCRL effectively generalizes and subsumes other more restricted formulations.

For one, as evident from the construction of PCRL, if we remove the context-specific components, $\mathcal{Z}$, $\mathcal{W}$ and $\mathbf{C}$, then PCRL collapses to the original Bayesian Poisson factorization (Cemgil, 2009; Gopalan et al., 2015) that would learn from user-item preferences alone.

For another, if we drop the collaborative filtering components, namely $\mathbf{X}$ and $\boldsymbol{\theta}$, then we would recover an instance of Deep Exponential Families (DEFs) (Ranganath et al., 2015) for unsupervised feature learning. However it should be noted that our composition of Gamma distributed layers and Gaussian weights has not been investigated previously in (Ranganath et al., 2015). The PCRL's representation learning component is also related to the Poisson Gamma Belief Network (PGBN) (Zhou et al., 2016). The key differences are: PGBN uses Dirichlet weights, it factorizes and chains the Gamma shape instead of the rate parameters.

If we further take the shape parameter λ_z^s to infinity, than PCRL is reduced to a Bayesian deep "decoder" neural network, with a stochastic Gamma top layer $\boldsymbol{\beta}$. Furthermore, starting from PCRL we can derive a Bayesian Gamma-Poisson variant of the variational auto-encoder (Kingma and Welling, 2014). To our knowledge, such neural networks with Gamma stochastic layers have not been studied in prior literature.

4 INFERENCE & LEARNING

So far we describe PCRL as a generative model. In practice, we are given $\mathbf{X}$ and $\mathbf{C}$, and we are interested in reversing the above generative process to infer the posterior distribution of the latent variables, i.e., $p(\boldsymbol{\theta}, \boldsymbol{\beta}, \mathcal{W} | \mathbf{X}, \mathbf{C})$ that would be the most likely to generate the observations. This allows us to explore data in different ways as well as predict unknown ratings for recommendations. Note that by taking λ_z^s to infinity the intermediate latent variables $\mathcal{Z}$ become deterministic; this is why they are not considered in the above posterior.

As in many Bayesian models, the above posterior is intractable. We therefore resort to approximate inference. In particular, we rely on Variational Inference (VI) (Bishop, 2006; Blei et al., 2017), which is widely used in statistical learning to fit complex Bayesian models.

4.1 THE VARIATIONAL FAMILY

The key to variational inference is to introduce a tractable family of distributions q, governed by a set of *variational parameters* ν. The objective is then to find the closest, typically in terms of the Kullback-Leibler (KL) divergence, member of this family to the true posterior.

We can ease inference in the collaborative part of PCRL by introducing an additional layer of auxiliary latent variables, leaving the original model intact when marginalized out. As in (Cemgil, 2009), we add K variables $s_{uik} \sim \text{Poisson}(\theta_{uk}\beta_{ik})$ for each observed rating x_{ui}, such that $x_{ui} = \sum_k s_{uik}$. The marginal distribution of x_{ui} is preserved thanks to the additive property of Poisson random variables (Kingman, 1993). As the s_{uik}'s are not random when x_{ui} is zero, we need to consider these variables for the non-zero elements in $\mathbf{X}$ only.

One main source of intractability in our model is the coupling between the different latent variables. To overcome this difficulty, we adopt a *mean-field* variational family (Jordan et al., 1999), $q(\cdot|\nu) = q(\theta, \beta, s, \mathcal{W}|\nu)$, which factorizes with respect to the latent variables:

$$q(\cdot|\nu, \mathbf{C}) = \prod_u q(\boldsymbol{\theta}_u|\tilde{\boldsymbol{\lambda}}_u^\theta) \prod_i q(\boldsymbol{\beta}_i|\tilde{\boldsymbol{\lambda}}_i^\beta) \prod_{u,i} q(\boldsymbol{s}_{ui}|\tilde{\boldsymbol{\phi}}_{ui}) \prod_{\ell=0}^L q(\mathbf{W}_\ell|\tilde{\boldsymbol{\xi}}_\ell) \quad (4)$$

where $\nu = \{\tilde{\lambda}, \tilde{\xi}, \tilde{\phi}\}$. Note that the variational distributions in the above equation are fully factorized, e.g., $q(\boldsymbol{\theta}_u|\tilde{\boldsymbol{\lambda}}_u^\theta) = \prod_k q(\theta_{uk}|\tilde{\boldsymbol{\lambda}}_{uk}^\theta)$. Each variational distribution is in the same family as the model distribution.

That is, the factors over the Gamma variables, $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$, are also Gamma distributions variational parameters $\boldsymbol{\lambda}$, e.g., $\tilde{\boldsymbol{\lambda}}^{\theta}_{uk} = (\tilde{\lambda}^{\theta,s}_{uk}, \tilde{\lambda}^{\theta,r}_{uk})$. For the item attributes, we further amortize computations by using an inference network. More precisely, we let $\tilde{\boldsymbol{\lambda}}^{\beta}_{i} = (\tilde{\boldsymbol{\lambda}}^{\beta,s}_{i}, \tilde{\boldsymbol{\lambda}}^{\beta,r}_{i}) = \mathtt{f}_{\omega}(\mathbf{c}_i)$, where $\mathtt{f}_{\omega}(\mathbf{c}_i)$ is a deep "encoder" neural network (MLP), parameterized by ω, whose input is $\mathbf{c}_i$, $\tilde{\boldsymbol{\lambda}}^{\beta,s}_{i} = (\tilde{\lambda}^{\beta,s}_{ik}, \dots, \tilde{\lambda}^{\beta,s}_{iK})$ and $\tilde{\boldsymbol{\lambda}}^{\beta,r}_{i} = (\tilde{\lambda}^{\beta,r}_{ik}, \dots, \tilde{\lambda}^{\beta,r}_{iK})$. Note that, the variational parameters over the item factors $q(\boldsymbol{\beta})$ become ω.

The factors over $\boldsymbol{s}_{ui}$ are Multinomial distributions with free parameters $\tilde{\boldsymbol{\phi}}$. This follows from the fact that the conditional distribution of a set of Poisson variables given their sum is a Multinomial (Cemgil, 2009).

The variational factor over $\mathbf{W}_{\ell}$ takes this form: $q(\mathbf{W}_{\ell}|\tilde{\boldsymbol{\xi}}_{\ell}) = \prod_{k=1}^{K_{\ell}} q(\mathbf{w}_{\ell,k}|\tilde{\boldsymbol{\xi}}^{k}_{\ell})$, where $\tilde{\boldsymbol{\xi}}^{k}_{\ell} = \{\tilde{\boldsymbol{\mu}}^{k}_{\ell}, (\tilde{\boldsymbol{\sigma}}^{k}_{\ell})^2\mathbf{I}_{K_{\ell+1}}\}$ indexes a multivariate Gaussian with a diagonal covariance structure.

Fitting the variational parameters ν by minimizing the KL divergence between q and the true posterior is akin to maximizing the Evidence Lower BOund (ELBO), i.e.,

$$\mathcal{L} = \mathbb{E}_{q}[\log p(\mathbf{X}, \mathbf{C}, \mathcal{W}, \boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{s}) - \log(q(\cdot|\nu))] \quad (5)$$

Next we derive an algorithm to maximize (5).

4.2 COORDINATE ASCENT LEARNING

We now derive a variational algorithm to estimate PCRL form data. The principle is to alternate the update of each variational parameter while holding the others fixed.

Updates for $\tilde{\boldsymbol{\lambda}}^{\theta}$ and $\tilde{\boldsymbol{\phi}}$. Thanks to the auxiliary variables $\boldsymbol{s}$, $\tilde{\boldsymbol{\lambda}}^{\theta}$ and $\tilde{\boldsymbol{\phi}}$ have the following closed-form updates,

$$\tilde{\lambda}^{\theta}_{uk} = \left(\lambda^{s}_{\theta} + \textstyle\sum_i x_{ui}\tilde{\phi}_{uik}, \lambda^{r}_{\theta} + \sum_i \frac{\tilde{\lambda}^{\beta,s}_{ik}}{\tilde{\lambda}^{\beta,r}_{ik}}\right), \quad (6)$$

$$\tilde{\phi}_{uik} \propto \exp\left(\psi(\tilde{\lambda}^{\theta,s}_{uk}) - \log\tilde{\lambda}^{\theta,r}_{uk} + \psi(\tilde{\lambda}^{\beta,s}_{ik}) - \log\tilde{\lambda}^{\beta,r}_{ik}\right) \quad (7)$$

where $\psi(\cdot)$ denotes the diagamma function. These updates are identical to those of Bayesian PF (Cemgil, 2009; Gopalan et al., 2015). For more details, please refer to the supplementary material (A.1).

Parameter update for $q(\boldsymbol{\beta})$ and $q(\mathcal{W})$. The remaining variational parameters do not admit closed-form updates. We therefore rely on stochastic steepest gradient ascent to optimize the ELBO according to these parameters.

Keeping only terms which are function of $\mathcal{W}$ or $\boldsymbol{\beta}$, the ELBO can be rewritten, for each item i, as follows

$$\mathcal{L}_i = \mathbb{E}_{q}[\log p(\boldsymbol{s}|\boldsymbol{\theta}, \boldsymbol{\beta}_i)] + \mathbb{E}_{q}[\log p(\mathbf{c}_i|\mathcal{W}, \boldsymbol{\beta}_i)] - \mathrm{KL}(q(\boldsymbol{\beta}_i)||p(\boldsymbol{\beta}_i)) - \mathrm{KL}(q(\mathcal{W})||p(\mathcal{W})) + \text{const} \quad (8)$$

with $\mathcal{L} = \sum_i \mathcal{L}_i$. While the first expectation and KL terms in (8) are available analytically, the second expectation over $\log p(\mathbf{c}_i|\mathcal{W}, \boldsymbol{\beta}_i)$ is intractable for general PCRL with respect to both $\mathcal{W}$ and $\boldsymbol{\beta}_i$. We cannot always push the expectations inward non-linear activation functions a_{ℓ}. This makes the direct evaluation of the gradient of $\mathcal{L}_i$ problematic. To overcome this difficulty we build a Monte Carlo estimator of the gradient of $\mathbb{E}_{q}[\log p(\mathbf{c}_i|\mathcal{W}, \boldsymbol{\beta}_i)]$. To this end, we rely on the recent Rejection Sampling Variational Inference (RSVI) method (Naesseth et al., 2017), which generalizes the *reparameterization trick* (Kingma and Welling, 2014; Rezende et al., 2014).

RSVI requires continuous latent variables, and its applicability depends on whether we can sample from the variational distribution $q(\beta; \omega)$ using the following reparameterization: (i) draw $\epsilon \sim \pi(\epsilon; \omega)$, (ii) $\beta = \mathcal{G}(\epsilon, \omega)$, where $\mathcal{G}$ is a deterministic function (mapping) that must be differentiable with respect to ω, and the distribution $\pi(\epsilon; \omega)$, defined by a rejection sampling algorithm, takes the following form,

$$\pi(\epsilon; \omega) = t(\epsilon)\frac{q(\mathcal{G}(\epsilon, \omega); \omega)}{r(\mathcal{G}(\epsilon, \omega); \omega)}, \quad (9)$$

where r and t are respectively the *proposal* and original distributions of ϵ used in rejection sampling. In this procedure, some samples from t are not valid (and therefore rejected), here we are interested in the distribution of the accepted samples $\pi(\epsilon; \omega)$. For more details, please refer to the supplementary material (A.2.1) where we provide a brief review of the reparametrized acceptance-rejection algorithm in our notations.

Assuming that we have a reparameterized acceptance-rejection sampling procedure to simulate from $q(\beta_{ik}; \omega)$, the next step is to rewrite $\mathbb{E}_{q(\boldsymbol{\beta}_i;\omega)}[\log p(\mathbf{c}_i|\mathcal{W}, \boldsymbol{\beta}_i)]$ as an expectation with respect to $\pi(\boldsymbol{\epsilon}_i; \omega)$ as follows

$$\mathbb{E}_{q(\boldsymbol{\beta}_i;\omega)}[\log p(\mathbf{c}_i|\mathcal{W}, \boldsymbol{\beta}_i)] = \mathbb{E}_{\pi(\boldsymbol{\epsilon}_i;\omega)}[\log p(\mathbf{c}_i|\mathcal{W}, \mathcal{G}(\boldsymbol{\epsilon}_i, \omega))] \quad (10)$$

where, $\boldsymbol{\epsilon}_i = \{\epsilon_{i1}, \dots, \epsilon_{iK}\}$, and π fully factorizes over the components of $\boldsymbol{\epsilon}_i$. The form of $\mathcal{G}(\boldsymbol{\epsilon}_i, \omega)$ will be given shortly. Based on (10) the gradient of $\mathbb{E}_{q(\boldsymbol{\beta}_i;\omega)}[\log p(\mathbf{c}_i|\mathcal{W}, \boldsymbol{\beta}_i)]$ is

$$\nabla_{\omega}\mathbb{E}_{q(\boldsymbol{\beta}_i;\omega)}[\log p(\mathbf{c}_i|\mathcal{W}, \boldsymbol{\beta}_i)] = \mathbb{E}_{\pi(\boldsymbol{\epsilon}_i;\omega)}[\log p(\mathbf{c}_i|\mathcal{W}, \mathcal{G}(\boldsymbol{\epsilon}_i, \omega))\nabla_{\omega}\log\pi(\boldsymbol{\epsilon}_i; \omega)] + \mathbb{E}_{\pi(\boldsymbol{\epsilon}_i;\omega)}[\nabla_{\omega}\log p(\mathbf{c}_i|\mathcal{W}, \mathcal{G}(\boldsymbol{\epsilon}_i, \omega))] \quad (11)$$

where we have pushed the gradient into the integral, used the log derivative-trick or REINFORCE (Glynn, 1990; Williams, 1992), and expressed integrals as expectations. All the derivations details of equations (11) and (10) are given in the supplementary material (A.2.2).

We can now form an unbiased Monte Carlo estimate of the above gradient as follows:

$$\nabla_\omega \mathbb{E}_{q(\boldsymbol{\beta}_i;\omega)}[\log p(\mathbf{c}_i|\mathcal{W},\boldsymbol{\beta}_i)] \simeq \frac{1}{M}\sum_{m=1}^{M} \log p(\mathbf{c}_i|\mathcal{W},\boldsymbol{\beta}_i^m)\nabla_\omega \log \frac{q(\mathcal{G}(\epsilon_i^m,\omega);\omega)}{r(\mathcal{G}(\epsilon_i^m,\omega);\omega)} + \frac{1}{M}\sum_{m=1}^{M}\nabla_\omega \log p(\mathbf{c}_i|\mathcal{W},\boldsymbol{\beta}_i^m) \quad (12)$$

where $\boldsymbol{\beta}_i^m = \{\beta_{i1}^m,\ldots,\beta_{iK}^m\}$, and $\beta_{ik}^m = \mathcal{G}(\epsilon_{ik}^m,\omega)$, with $\epsilon_{ik}^m \sim \pi(\epsilon_{ik},\omega)$. In practice we set $M=1$.

Following Naesseth et al. (2017), for the Gamma random variables, we use the reparameterization proposed by Marsaglia and Tsang (2000). For a Gamma$(\lambda_\omega^s,\lambda_\omega^r)$, such that $\lambda_\omega^s \geq 1$, we use:

$$\mathcal{G}(\epsilon,\omega) = \frac{1}{\lambda_\omega^r}\left(\lambda_\omega^s - \frac{1}{3}\right)\left(1+\frac{\epsilon}{\sqrt{9\lambda_\omega^s-3}}\right) \quad (13)$$

with $\epsilon \sim t(\epsilon) = \mathcal{N}(0,1)$. When the shape parameter is less than 1, $\lambda_\omega^s < 1$, we use the shape augmentation technique (Marsaglia and Tsang, 2000). That is, if $\tilde{\beta} \sim$ Gamma(λ^s+1,λ^r), and $\beta = u^{\frac{1}{\lambda^s}}\tilde{\beta}$ with $u \sim \mathcal{U}[0,1]$, then $\beta \sim$ Gamma(λ^s,λ^r).

Approximating the gradient of the ELBO with respect to $\tilde{\boldsymbol{\xi}}$ is simpler since the Gaussian satisfies the requirements of the original reparameterization trick (Kingma and Welling, 2014; Rezende et al., 2014). Roughly, the second expectation in (11) vanishes since the marginal distribution of the samples ϵ is independent of the variational parameters $\tilde{\xi}$. Hence, the Monte Carlo estimator of $\nabla_{\tilde{\xi}}\mathbb{E}_{q(\mathcal{W})}[\log p(\mathbf{c}_i|\mathcal{W},\boldsymbol{\beta}_i)]$ takes this form:

$$\nabla_{\tilde{\xi}}\mathbb{E}_{q(\mathcal{W};\tilde{\xi})}[\log p(\mathbf{c}_i|\mathcal{W},\boldsymbol{\beta}_i)] \simeq \frac{1}{M}\sum_{m=1}^{M}\nabla_{\tilde{\xi}}\log p(\mathbf{c}_i|\mathcal{W}^m,\boldsymbol{\beta}_i) \quad (14)$$

where $\mathcal{W}^m = \{\mathbf{W}_1^m,\ldots,\mathbf{W}_L^m\}$, $\mathbf{w}_{\ell,k}^m = \mathcal{T}(\boldsymbol{\eta}^m,\tilde{\boldsymbol{\xi}}_\ell^k) = \tilde{\boldsymbol{\mu}}_\ell^k + \tilde{\boldsymbol{\sigma}}_\ell^k \odot \boldsymbol{\eta}^m$, and $\boldsymbol{\eta}^m \sim \mathcal{N}(\mathbf{0},\mathbf{I})$, the notation $\odot$ refers to the Hadamard product.

Putting it all together, our Monte Carlo estimator for the gradient of the ELBO, is given by:

$$\nabla_{\omega,\tilde{\xi}}\mathcal{L}_i \simeq I\nabla_\omega\left(\mathbb{E}_q[\log p(\boldsymbol{s}|\boldsymbol{\theta},\boldsymbol{\beta}_i)] - \mathrm{KL}(q(\boldsymbol{\beta}_i)||p(\boldsymbol{\beta}_i))\right) + \frac{I}{M}\sum_{m=1}^{M}\nabla_{\omega,\tilde{\xi}}\log p(\mathbf{c}_i|\mathcal{W}^m,\boldsymbol{\beta}_i^m) + \frac{I}{M}\sum_{m=1}^{M}\log p(\mathbf{c}_i|\mathcal{W}^m,\boldsymbol{\beta}_i^m)\nabla_\omega \log\frac{q(\mathcal{G}(\epsilon_i^m,\omega);\omega)}{r(\mathcal{G}(\epsilon_i^m,\omega);\omega)} - \nabla_{\tilde{\xi}}\mathrm{KL}(q(\mathcal{W})||p(\mathcal{W})) \quad (15)$$

With the estimator (15) in place, we perform stochastic gradient ascent over the parameters ω and $\tilde{\xi}$. We use backpropagation to evaluate the gradients over the weights of the inference network ω. In particular, we use RMSProp to scale the gradients before applying them. In practice, we take several stochastic gradient steps to nearly optimize the ELBO with respect to ω and $\tilde{\xi}$, before to perform coordinate ascent step to update $\tilde{\boldsymbol{\lambda}}^\theta$ and $\tilde{\boldsymbol{\phi}}$. More precisely, after each epoch of stochastic gradient ascent we update $\tilde{\boldsymbol{\lambda}}^\theta$ and $\tilde{\boldsymbol{\phi}}$.

4.3 MISSING RATINGS ESTIMATION

Once PCRL is fit to the obsevations, we can estimate the unknown ratings for each user u and item i as follows

$$\hat{x}_{ui} = \mathbb{E}_q(\boldsymbol{\theta}_u^\top\boldsymbol{\beta}_i) = \mathbb{E}_q(\boldsymbol{\theta}_u)^\top\mathbb{E}_q(\boldsymbol{\beta}_i), \quad (16)$$

Note that this expectation in intractable with respect to the true posterior. These predicted values are then used to rank unrated items for each user so as to provide her with a recommendation list.

4.4 DESIRABLE PROPERTIES

The variational PCRL enjoys several desirable properties. In terms of efficiency, the operations involving user-item and item-context interactions need to be carried out only for the non-zero entries in $\mathbf{X}$ and $\mathbf{C}$. It can be shown that the computational time complexity of the variational PCRL algorithm (its batch version) is linear in the number of non-zeros entries in $\mathbf{X}$ and $\mathbf{C}$.

The main intuition behind PCRL is to learn item representations encoding various contextual regularities among items that are good at explaining the user behaviour. Interestingly, this intuition is reflected theoretically, as seen in the proposition below. Note that this result arises naturally from our formulation.

Proposition 1 *Let $q(\boldsymbol{\beta}_i;\omega)$ be the variational distribution over the item factor in PCRL. Then, for fixed $\tilde{\boldsymbol{\lambda}}^\theta$, $\tilde{\boldsymbol{\phi}}$ and $\tilde{\boldsymbol{\xi}}$, maximizing the ELBO (5) with respect to ω is equivalent to maximizing the following criterion:*

$$\textstyle\sum_i \mathbb{E}_q[\log p(\mathbf{c}_i|\mathcal{W},\boldsymbol{\beta}_i)] - KL(q(\boldsymbol{\beta}_i;\omega)||\tilde{q}(\boldsymbol{\beta}_i)). \quad (17)$$

where $\tilde{q}(\boldsymbol{\beta}_i)$ denotes the optimal mean-field varitional distribution over the item attributes in Bayesian Poisson factorization. That is, $\tilde{q}(\boldsymbol{\beta}_i) = \prod_k \tilde{q}(\beta_{ik})$, and $\tilde{q}(\beta_{ik}) = Gamma(\lambda_\beta^s + \sum_u x_{ui}\tilde{\phi}_{uik}, \lambda_\beta^r + \sum_u \frac{\tilde{\lambda}_{uk}^s}{\tilde{\lambda}_{uk}^r})$.

The proof is given below. The KL term in the above proposition can be viewed as a regularizer which encourages PCRL's variational factor over the items, $q(\boldsymbol{\beta}_i;\omega)$ to look like its optimal mean-field counterpart in Bayesian

Poisson factorization $\tilde{q}(\boldsymbol{\beta}_i)$. Recall that $\tilde{q}(\boldsymbol{\beta}_i)$ is independent of the item context $\mathbf{C}$, and puts high probability on configurations of $\boldsymbol{\beta}_i$ that explain user preferences. This makes it clear how the collaborative PF component in PCRL guides or encourages the representation learning part to focus on extracting contextual features that might be useful for explaining user preferences. From this perspective, PCRL can be interpreted as regularizing a deep generative model with Bayesian Poisson Factorization.

Proof. If we fix all the variational parameters except ω, then maximizing the ELBO with respect to the latter is equivalent to maximizing

$$\mathcal{L}_i = \mathbb{E}_q[\log p(\boldsymbol{s}|\boldsymbol{\theta}, \boldsymbol{\beta}_i) + \log p(\boldsymbol{\beta}_i)] + \mathbb{E}_q[\log p(\mathbf{c}_i|\mathcal{W}, \boldsymbol{\beta}_i) - \log q(\boldsymbol{\beta}_i; \omega)] + \text{const.} \quad (18)$$

In particular, we have

$$\log p(\boldsymbol{\beta}_{ik}) \propto (\lambda^s_\beta - 1)\log(\beta_{ik}) - \lambda^r_\beta \beta_{ik}, \text{ and,}$$

$$\log p(s_{uik}|\theta_{uk}, \boldsymbol{\beta}_{ik}) \propto s_{uik}\log(\beta_{ik}) - \theta_{uk}\beta_{ik}.$$

Therefore we get

$$\mathbb{E}_{q(\theta,s)}[\log p(\boldsymbol{s}|\boldsymbol{\theta}, \boldsymbol{\beta}_i) + \log p(\boldsymbol{\beta}_i)] = -(\lambda^r_\beta + \sum_u \frac{\tilde{\lambda}^s_{uk}}{\tilde{\lambda}^r_{uk}})\beta_{ik} + (\lambda^s_\beta + \sum_u x_{ui}\tilde{\phi}_{uik} - 1)\log\beta_{ik} + \text{const}, \quad (19)$$

where we recognize the log (up to the normalizing constant) of the following Gamma$(\lambda^s_\beta + \sum_u x_{ui}\tilde{\phi}_{uik}, \lambda^r_\beta + \sum_u \frac{\tilde{\lambda}^s_{uk}}{\tilde{\lambda}^r_{uk}})$ distribution. Adding the normalizing constant (which is independent of ω) and plugin (19) into (18), completes the proof.■

5 EXPERIMENTS

In this section, we study the impact of item context, and our modeling assumptions, on personalized item recommendation as well as item representation learning.

Datasets. We use five datasets from `Amazon.com`[3], provided by McAuley et al. (2015b,a). They include both user-item interactions and the "Also Viewed" lists that we treat as item contexts. We preprocess all datasets so that each user (resp. item) has at least ten (resp. two) ratings, and the sets of row and column items in $\mathbf{C}$ are identical. Table 1 reports the resulting statistics.

Comparative Models. We benchmark `PCRL`[4] against strong comparable generative factorization models.

- `MCF`: Matrix Co-Factorization (Park et al., 2017) incorporates item-item relationships into Gaussian MF.
- `PF`: Bayesian Poisson Factorization (Gopalan et al., 2015) arises as a special case of our model without the context-specific components. Comparison to `PF` allows us to assess the impact of item contexts.
- `CTPF`: Collaborative Topic Poisson Factorization (Gopalan et al., 2014b) was developed for content-based recommendation, but can serve as baseline by substituting item-word matrix with item-context $\mathbf{C}$.
- `CoCTPF`: Content-only `CTPF` (Gopalan et al., 2014b) is a variant of `CTPF` without the document topic offsets; please refer to (Gopalan et al., 2014b) for details. Comparison to `CoCTPF` allows us to assess the impact of our modeling choice of multilayered representation learning, as opposed to PF, for item context.
- `RL+PF`: Representation Learning + PF is a two-stage pipelined approach, which models item context independently from user preferences. First, it infers $q(\beta)$ from $\mathbf{C}$ using PCRL's representation learning-spesifc part. Second, it performs `PF` on $\mathbf{X}$ to infer $q(\theta)$ while holding the item factors fixed. Comparison to `RL+PF` allows us to assess the benefit of our unified modeling.

Datasets	Characteristics					
	#Users	#Items	#Ratings	nz_X (%)	$\#nz_C$	nz_C (%)
Office	3,703	6,523	53,282	0.22	108,466	0.25
Grocery	8,938	22,890	148,735	0.07	480,300	0.09
Automotive	7,280	15,635	63,477	0.05	365,634	0.15
Sports	19,049	24,095	211,582	0.04	531,148	0.09
Pet Supplies	16,462	20,049	164,017	0.05	631,102	0.16

Table 1: Statistics of the Datasets.

Experimental Setup. For each dataset, we randomly select 80% of the ratings as training data and the remaining 20% as test data. Random selection is carried out three times independently on each dataset. The reported result is the average performance over the three samples.

Following previous works (Gopalan et al., 2014a, 2015), we set the number of latent dimensions for user preferences θ and item attributes β to 100. In all experiments, we use a two-layer PCRL $(\mathbf{z}_1, \boldsymbol{\beta})$ with dimensions $(100, 300)$ in the item representation learning component. The activation functions at the layers $(\mathbf{c}, \mathbf{z}_1)$ are set to (sigmoid, relu). Similarly, we use a two-layer inference network (encoder) with dimensions $(300, 100 + 100)$—recall that this network outputs Gamma variational parameters, a total of 100 (shape) + 100 (rate) parameters—and activation functions (relu, sofplus). When necessary we add a small offset to ensure strict positivity, e.g., the rate of the Poisson, the shape and rate of the Gamma, all must be positive. To encourage sparse latent representations, we set Gamma prior parameters (λ^s, λ^r) to $(0.3, 0.3)$—resulting in exponen-

[3] http://jmcauley.ucsd.edu/data/amazon/

[4] source code available at: https://cornac.preferred.ai/

Table 2: Average recommendation accuracy.

	Metric	MCF	PF	CTPF	CoCTPF	RL+PF	PCRL
Office Prod.	nDCG	0.1525	0.1663	0.1718	0.1806	0.1551	**0.1974**
	MRR	0.0239	0.0414	0.0467	0.0558	0.0237	**0.0708**
	Pre@20	0.0041	0.0096	0.0111	0.0129	0.0048	**0.0156**
	Rec@20	0.0293	0.0541	0.0615	0.0768	0.0325	**0.0873**
	Pre50	0.0033	0.0077	0.0075	0.0095	0.0039	**0.0116**
	Rec50	0.0569	0.0970	0.1021	0.1392	0.0654	**0.1627**
Grocery	nDCG	0.1286	0.1568	0.1553	0.1717	0.1295	**0.1801**
	MRR	0.0145	0.0452	0.0429	0.0529	0.0098	**0.0652**
	Pre20	0.0024	0.0095	0.0095	0.0116	0.0017	**0.0134**
	Rec20	0.0191	0.0571	0.0591	0.0739	0.0109	**0.0751**
	Pre50	0.0019	0.0070	0.0072	0.0086	0.0015	**0.0098**
	Rec50	0.0353	0.1021	0.1090	0.1213	0.0234	**0.1339**
Automotive	nDCG	0.1186	0.1123	0.1124	0.1417	0.1225	**0.1453**
	MRR	0.0121	0.0100	0.0103	0.0337	0.0111	**0.0350**
	Pre20	0.0022	0.0015	0.0016	0.0058	0.0017	**0.0063**
	Rec20	0.0228	0.0132	0.0143	**0.0566**	0.0147	0.0536
	Pre50	0.0016	0.0010	0.0012	0.0038	0.0016	**0.0043**
	Rec50	0.0393	0.0233	0.0262	**0.0920**	0.0325	0.0913
Sports	nDCG	0.1122	0.1179	0.1189	0.1398	0.1190	**0.1524**
	MRR	0.0071	0.0122	0.0119	0.0297	0.0073	**0.0375**
	Pre20	0.0011	0.0018	0.0022	0.0054	0.0014	**0.0070**
	Rec20	0.0096	0.0143	0.0170	0.0431	0.0113	**0.0507**
	Pre50	0.0009	0.0013	0.0017	0.0038	0.0013	**0.0051**
	Rec50	0.0192	0.0273	0.0318	0.0759	0.0298	**0.0942**
Pet Supplies	nDCG	0.1201	0.1288	0.1317	0.1585	0.1210	**0.1626**
	MRR	0.0136	0.0207	0.0237	0.0441	0.0094	**0.0461**
	Pre20	0.0022	0.0029	0.0034	0.0079	0.0019	**0.0088**
	Rec20	0.0237	0.0271	0.0314	0.0752	0.0167	**0.0776**
	Pre50	0.0016	0.0021	0.0028	0.0055	0.0016	**0.0063**
	Rec50	0.0397	0.0481	0.0561	0.1301	0.0359	**0.1455**

tially shaped Gamma distributions with mean equal to 1. For an illustration, please refer to Figure 2 in (Cemgil, 2009). We follow the same strategy, grid search, as in (Park et al., 2017) to set the different hyperparameters of MCF. In order for the comparisons to be fair, we use the same random parameters to initialize all PF-based models, where it is possible.

Item Recommendation. Here we look into the quality of item recommendation, and discuss item representation later. We assess the recommendation accuracy on a set of held-out items—the test set. We retain four widely used measures for top-M recommendation, namely the Normalized Discount Cumulative Gain (nDCG), Mean Reciprocal Rank (MRR), Precision@M (P@M) and Recall@M (R@M), where M is the number of items in the recommendation list (Bobadilla et al., 2013). Intuitively, nDCG and MRR measures the ranking quality of a model, while Precision@M and Recall@M assess the quality of a user's top-M recommendation list. These measures vary from 0.0 to 1.0 (higher is better).

Table 2 depicts the average performances[5] of the competing models in terms of different metrics, over all datasets. For the sake of completeness we also report, in Table 3, the average log-likelihood values for the Poisson models, i.e., $\log p(\mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\beta})$, where we set $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ to their mean values under the corresponding variational distribution.

[5]Most of the standard deviation values are of order 1e-3/1e-4, we do not report them to fit Table 2 into one column.

Table 3: Comparison of Poisson log-likelihood.

Models	Office Prod.	Grocery	Automotive	Sports	Pet Supplies
PF	-210522	-680546	-355671	-1187849	-838712
CTPF	-208633	-681832	-354239	-1180927	-838910
CoCTPF	-207840	-656676	-336319	-1138744	-786326
RL+PF	-227454	-761403	-341730	-1178502	-887624
PCRL	**-199066**	**-649054**	**-322889**	**-1061088**	**-760935**

The main points from these results are as follows.

Item context is useful for personalized recommendation. The proposed PCRL substantially outperforms the other competing models in virtually all cases. In particular, the major difference between the original PF and our proposed PCRL as well as CTPF or CoTPF is that the latter models incorporate item context. We can therefore attribute the performance improvements reached by those over PF to the modeling of the item context.

Poisson Factorization performs better than its Gaussian counterpart. Effectively CoCTPF is the closest Poisson alternative to the Gaussian MCF. The former outperforms the latter in all cases. Even when augmented with contextual item information, the Poisson remains a better alternative than the Gaussian for modeling user preferences, which is in line with the findings of previous work on PF.

The hierarchical (multilayered) structure in PCRL is useful. The model PCRL can be viewed as an alternative to CoCTPF, where a multilayered generative model is substituted for PF to model item contexts. From Tables 2 and 3, we note that PCRL substantially outperforms CoCTPF on almost all datasets and across all metrics, except in terms of recall on Automotive. Since the main difference between the two approaches lies in how they model item context, these results suggest that our multilayered architecture does a better job than PF in extracting latent features from item's contexts.

Joint modeling or learning is beneficial. A key point to PCRL is to model user preferences and item contexts jointly. As Tables 2 and 3 show, PCRL outperforms the two-stage pipelined model RL+PF. Quite surprisingly, the latter performs even worse than PF on almost all datasets. This demonstrates the importance of joint modeling, and suggests that the PCRL's collaborative component plays an important role in guiding item representation learning towards extracting contextual features that are relevant for explaining user preferences. Whereas modeling the item context independently yields item representations that capture other item aspects, which are not necessarily as good for predicting user preferences.

To gain further insight into the results, especially the latter two points above, we conduct another series of experiments where we compare the quality of the item representations produced by the different models.

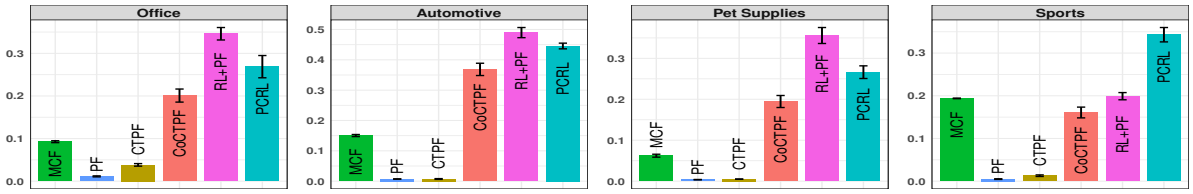

Figure 2: Average NMI over different datasets.

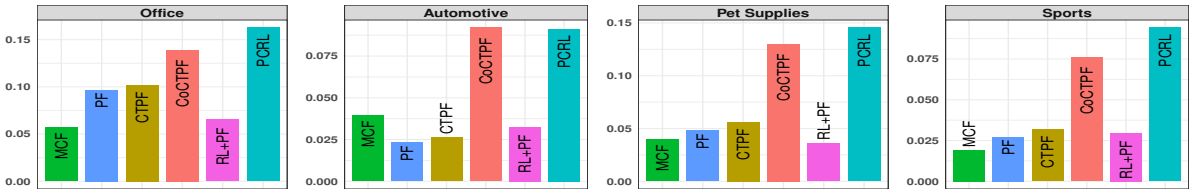

Figure 3: Average Recall@50 over different datasets.

Item Representations. Evaluating the quality of item representations is a challenging task. Here, we propose to make such an evaluation in terms of clustering. We seek to assess how well the representations produced by each model are good at organizing items into meaningful clusters. As evaluation measure, we use Normalized Mutual Information (NMI) (Strehl and Ghosh, 2002). Intuitively, NMI quantifies how much the estimated clustering is informative about the true clustering. As the "true" clustering, we retain the ten most frequent item categories (classes) in each datasest; these categories per dataset are listed in the supplementary material (B). We do not consider Grocery in this experiment, since its category labels are not available.

To form clusters based on learned item representations, we use the spherical k-means (`Skmeans`) (Dhillon and Modha, 2001). We perform fifty runs of (`Skmeans`), with different initial random points, and report the average NMI of the ten best runs—in terms of the Skmeans' criterion—as the final results. The fifty random starting points used by `Skmeans` are the same across all models.

Figure 2 reports the clustering results. For reference, Figure 3 reproduces the Recall@50 (the results are consistent across all metrics) on the item recommendation task.

`PF` that relies solely on user-item interactions obtains the worst clustering results. Such sparse information is not rich enough to allow `PF` infer relationships among items. The other models that use contextual information perform better. In particular, we note that `PCRL` produces representations that are better suited to organize items into categories than the `CoCTPF` models. This provides additional empirical support for the importance of our hierarchical architecture to model items' contexts.

Interestingly, `RL+PF` performs relatively well on clustering (Figure 2) even as it performs rather poorly on recommendation (Figure 3). One possible explanation of this phenomenon is that `RL+PF` focuses on item similarity. While this is beneficial for clustering, this might not always be useful for recommendation. Hypothetically, two similar items may be alternatives. Instead of recommending alternatives to an item that a user has purchased, it may be useful to recommend complementary items (which may not belong to the same category).

6 DISCUSSION

PCRL composes Bayesian Poisson factorization with a multilayered latent variable model to join both sources of data: user preferences and item contexts. Empirical results provide strong support for the benefits of our modeling framework and reflect the underlying assumption in PCRL, namely: the collaborative component guides the item representation learning towards extracting contextual features that are useful for the recommendation task, whereas the representation learning encourages the collaborative part to rely on item's contexts to explain recommendations, alleviating data sparsity.

While our focus here has been on item context, PCRL could potentially be extended to learn item representations from other modalities, e.g., text, images, etc. Another interesting direction of future work, is to investigate deeper variants of PCRL which would improve the feature learning.

We make PCRL's implementation publicly available as part of the CORNAC[6] recommendation library.

Acknowledgments

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its NRF Fellowship Programme (Award No. NRF-NRFF2016-07).

[6] https://cornac.preferred.ai/

References

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE TPAMI*, 35(8):1798–1828.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *JASA*.

Bobadilla, J., Ortega, F., Hernando, A., and Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-based systems*, 46:109–132.

Canny, J. (2004). Gap: a factor model for discrete data. In *SIGIR*, pages 122–129.

Cemgil, A. T. (2009). Bayesian inference for nonnegative matrix factorisation models. *Computational Intelligence and Neuroscience*, 2009.

Chaney, A. J., Blei, D. M., and Eliassi-Rad, T. (2015). A probabilistic model for using social networks in personalized item recommendation. In *RecSys*, pages 43–50.

Charlin, L., Ranganath, R., McInerney, J., and Blei, D. M. (2015). Dynamic poisson factorization. In *RecSys*, pages 155–162.

Dhillon, I. S. and Modha, D. S. (2001). Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1-2):143–175.

Glynn, P. W. (1990). Likelihood ratio gradient estimation for stochastic systems. *CACM*, 33(10):75–84.

Gopalan, P., Hofman, J. M., and Blei, D. M. (2015). Scalable recommendation with hierarchical poisson factorization. In *UAI*, pages 326–335.

Gopalan, P., Ruiz, F. J., Ranganath, R., and Blei, D. (2014a). Bayesian nonparametric poisson factorization for recommendation systems. In *Artificial Intelligence and Statistics*, pages 275–283.

Gopalan, P. K., Charlin, L., and Blei, D. (2014b). Content-based recommendations with poisson factorization. In *NIPS*, pages 3176–3184.

Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. *ICLR*.

Kingman, J. F. C. (1993). *Poisson processes*. Wiley Online Library.

Koenigstein, N., Dror, G., and Koren, Y. (2011). Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy. In *RecSys*, pages 165–172.

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

Li, X. and She, J. (2017). Collaborative variational autoencoder for recommender systems. In *KDD*, pages 305–314.

Liang, D., Altosaar, J., Charlin, L., and Blei, D. M. (2016). Factorization meets the item embedding: Regularizing matrix factorization with item cooccurrence. In *RecSys*, pages 59–66.

Ma, H., Yang, H., Lyu, M. R., and King, I. (2008). Sorec: social recommendation using probabilistic matrix factorization. In *CIKM*, pages 931–940.

Marsaglia, G. and Tsang, W. W. (2000). A simple method for generating gamma variables. *TOMS*, 26(3):363–372.

McAuley, J., Pandey, R., and Leskovec, J. (2015a). Inferring networks of substitutable and complementary products. In *KDD*, pages 785–794.

McAuley, J., Targett, C., Shi, Q., and Van Den Hengel, A. (2015b). Image-based recommendations on styles and substitutes. In *SIGIR*, pages 43–52.

Mnih, A. and Salakhutdinov, R. R. (2008). Probabilistic matrix factorization. In *NIPS*, pages 1257–1264.

Naesseth, C., Ruiz, F., Linderman, S., and Blei, D. (2017). Reparameterization gradients through acceptance-rejection sampling algorithms. In *AISTATS*, pages 489–498.

Park, C., Kim, D., Oh, J., and Yu, H. (2017). Do also-viewed products help user rating prediction? In *WWW*, pages 1113–1122.

Park, Y.-J. and Tuzhilin, A. (2008). The long tail of recommender systems and how to leverage it. In *RecSys*, pages 11–18. ACM.

Ranganath, R., Tang, L., Charlin, L., and Blei, D. (2015). Deep exponential families. In *AISTATS*, pages 762–771.

Rao, N., Yu, H.-F., Ravikumar, P. K., and Dhillon, I. S. (2015). Collaborative filtering with graph information: Consistency and scalable methods. In *NIPS*, pages 2107–2115.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, pages 1278–1286.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295. ACM.

Shan, H. and Banerjee, A. (2010). Generalized probabilistic matrix factorizations for collaborative filtering. In *ICDM*, pages 1025–1030.

Singh, A. P. and Gordon, G. J. (2008). Relational learning via collective matrix factorization. In *KDD*, pages 650–658.

Strehl, A. and Ghosh, J. (2002). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *JMLR*, 3:583–617.

Wang, C. and Blei, D. M. (2011). Collaborative topic modeling for recommending scientific articles. In *KDD*, pages 448–456.

Wang, H., Wang, N., and Yeung, D.-Y. (2015). Collaborative deep learning for recommender systems. In *KDD*, pages 1235–1244.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer.

Zhou, M., Cong, Y., and Chen, B. (2016). Augmentable gamma belief networks. *Journal of Machine Learning Research*, 17(163):1–44.

Zhou, T., Shan, H., Banerjee, A., and Sapiro, G. (2012). Kernelized probabilistic matrix factorization: Exploiting graphs and side information. In *SDM*, pages 403–414.