NIPS 2014 Workshop on *Personalization: Methods and Applications*. Montreal, Canada

A Scalable Bayesian Alternative to Density Estimation with a Bilinear Softmax Function

Ulrich Paquet
Microsoft Research
Cambridge, United Kingdom

Noam Koenigstein
Microsoft R&D
Herzliya, Israel

Ole Winther
Technical University of Denmark
Lyngby, Denmark

Abstract

We present a novel, scalable and Bayesian approach to modelling the occurrence of pairs (i, j) drawn from a large vocabulary. Our practical interest is in modelling $(\textit{user}, \textit{item})$ pairs in a recommender system, for which we present state of the art results on Xbox movie viewing data. The observed pairs are assumed to be generated by a simple popularity based selection process followed by censoring using a preference function. By basing inference on the well-founded principle of variational bounding, and using new site-independent bounds, we show how a scalable inference procedure can be obtained for large data sets. The model is a plausible alternative to modelling discrete densities with a bilinear softmax function.

1 Introduction

We present an new model for the occurrence of pairs of discrete symbols (i, j) from a finite set, which can be used to predict the occurrence of symbol j given that the other symbol is i. These pairs might be tuples of $(\textit{user}, \textit{item})$ purchase events, or a stream of $(\textit{user}, \textit{game})$ gameplay events. From such a model, a recommender system can be tailored around the conditional probability of item or game j, given user i. Alternatively, these tuples might be $(\textit{word}_1, \textit{word}_2)$ bigrams in a simple language model.

If there are I and J of each symbol, and $\mathbf{u}_i, \mathbf{v}_j \in \mathbb{R}^K$ (where $K \ll I, J$) are associated with user i and item j, an "ideal" model for the density of i and j is the bilinear softmax

$$p(i,j) = \mathrm{e}^{\mathbf{u}_i^T \mathbf{v}_j} \Big/ \sum_{i',j'} \mathrm{e}^{\mathbf{u}_{i'}^T \mathbf{v}_{j'}} \ . \tag{1}$$

Its normalizing constant sums over all $I \times J$ discrete options. When i is given, $p(j|i)$ defines softmax regression, the multi-class extension of logistic regression. The bilinear softmax function poses a practical difficulty: the large sums from the normalizing constant appear in the likelihood gradient through

$$\frac{\partial \log p(i,j)}{\partial \mathbf{u}_i} = \mathbf{v}_j - \sum_{k=1}^{J} \left(\frac{\mathrm{e}^{\mathbf{u}_i^T \mathbf{v}_k}}{\sum_{i',j'} \mathrm{e}^{\mathbf{u}_{i'}^T \mathbf{v}_{j'}}} \right) \mathbf{v}_k = \mathbf{v}_j - \sum_{j'=1}^{J} w_{ij'} \mathbf{v}_{j'} \ , \tag{2}$$

which requires a sum over all IJ pairs in its normalizer. These kinds of models appear often in neural probabilistic language models, and gradients are estimated with noise contrastive estimation [4, 8] or importance sampling [1]. The conditional density $p(j|i)$ can also be redefined as a tree-based hierarchy of smaller softmax functions [9]. Alternatively, modelling can be done by formulating a simpler objective function based on a classification likelihood, and including stochastically "negative sampled" pairs during optimization. This was done for skip-gram models that consider $(\textit{word}_1, \textit{word}_2)$ pairs [7] and for $(\textit{user}, \textit{item})$ pairs [10].

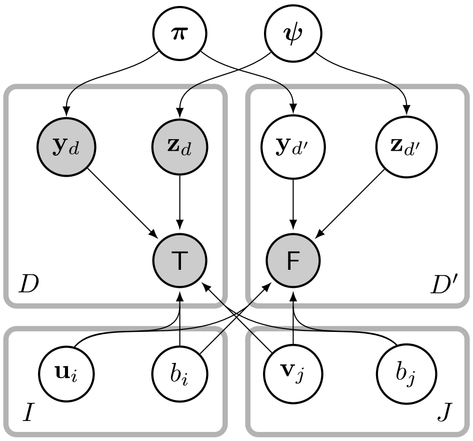

Figure 1: A generative model for observing D pairs of symbols, assuming that D' *unknown* pairs were censored.

1.1 An alternative view

Is it practically feasible to go beyond a point estimate and estimate parameter uncertainty when $I \approx 10^7$ and $J \approx 10^7$? Can we tractably estimate posterior uncertainty in the regime where data is scarce? In an recommender system with millions of users and items, we still have to reason about users with only one to a few interactions.

Equations (1) and (2) have pleasing properties: (1) embeds each i and j in a K-dimensional space, while (2) adjusts the embedding by pulling $\mathbf{u}_i$ towards $\mathbf{v}_j$ and pushes it further from all other $\mathbf{v}_{j'}$ when a pair (i, j) is observed. In the rest of this paper, we carefully construct a model that attempts to bridge the gap to (1), so that a variational Bayesian algorithm can be derived which does *not* scale with $I \times J$, as (2) does, but still keeps properties of (1) and (2).

On observing D pairs, the model is specified by a parameter D', the number of "censored" pairs, and its choice depends on D. At the expense of a slightly unnatural generative model, a scalable and computationally tractable approximate inference algorithm can be derived.[1]

2 Generative model for pairs with censoring

A pair (i, j) will be represented as a pair $(\mathbf{y}, \mathbf{z})$ of binary indicator vectors, where only bits i and j are "on" in $\mathbf{y} \in \{0, 1\}^I$ and $\mathbf{z} \in \{0, 1\}^J$ respectively. We shall model the data set by appending a binary variable $o = \mathsf{T}$ (true) to each pair: we *did* observe that symbols i and j co-occurred, user i played game j today, and so on. We therefore observe a data set of D pairs, which takes the form $\{o_d = \mathsf{T}, \mathbf{y}_d, \mathbf{z}_d\}_{d=1}^{D}$.

The censored approach assumes that there were a number of pairs that did not surface in the data set, such that $o = \mathsf{F}$ (false). *We do not know which pairs and how many they were*, but in practice we will allow the size of the censored set be specified as a hyperparameter D', and assume that $\{o_{d'} = \mathsf{F}\}_{d'=1}^{D'}$ is additionally provided. Let data $\mathscr{D} \doteq \{\{o_d = \mathsf{T}, \mathbf{y}_d, \mathbf{z}_d\}_{d=1}^{D}, \{o_{d'} = \mathsf{F}\}_{d'=1}^{D'}\}$ denote all observations. The ratio D/D' can be seen as a pre-specified positive to negative class ratio; various settings of r in $D' = rD$ are investigated in Sec. 4. The censored set constitutes a "negative background" against which the energy $\mathbf{u}_i^T \mathbf{v}_j$ will be fit, and it plays a role similar to that of the softmax normalizer in the gradient of $\log p(i, j)$ from (1): on observing a pair (i, j), $\mathbf{u}_i$ is pulled towards $\mathbf{v}_j$ and pushed further from all other $\mathbf{v}_{j'}$.

As a bridge towards (1), we propose a model which combines popularity-based selection with a personalized preference function to model (i, j):

[1] Detailed derivations follow in [11].

1. In a *selection step* a user i is chosen with probability π_i, and an item j is chosen with probability ψ_j.
2. In a *censoring step* the pair (i, j) is observed with probability $\sigma(\mathbf{u}_i^T \mathbf{v}_j + b_i + b_j)$ and censored with probability $1 - \sigma(\mathbf{u}_i^T \mathbf{v}_j + b_i + b_j)$, where $\sigma(a) = 1/(1 + e^{-a})$ is the logistic function.

The generative process is illustrated in Fig. 1, and is as follows: draw parameters $\boldsymbol{\vartheta}$ from their prior distributions (given explicitly below). Repeat drawing pairs (i, j) with indexes drawn from $\mathrm{Discrete}(\boldsymbol{\pi})$ and $\mathrm{Discrete}(\boldsymbol{\psi})$ and observe the pairs with probability $\sigma(\mathbf{u}_i^T \mathbf{v}_j + b_i + b_j)$. D such pairs are seen, while we assume that D', the number of censored data points, is specified as a hyperparameter.

Fixing notation, let $\mathbf{U} \doteq \{\mathbf{u}_i\}_{i=1}^{I}$ and $\mathbf{V} \doteq \{\mathbf{v}_j\}_{j=1}^{J}$ denote all bilinear parameters and $\mathbf{b} \doteq \{\{b_i\}_{i=1}^{I}, \{b_j\}_{j=1}^{J}\}$ denote biases, with $\boldsymbol{\vartheta} \doteq \{\mathbf{U}, \mathbf{V}, \mathbf{b}, \boldsymbol{\pi}, \boldsymbol{\psi}\}$. The density of an uncensored data point d is therefore

$$\begin{aligned} p(o_d = \mathsf{T}, \mathbf{y}_d, \mathbf{z}_d | \boldsymbol{\vartheta}) &= p(o_d = \mathsf{T} | \mathbf{y}_d, \mathbf{z}_d, \mathbf{U}, \mathbf{V}, \mathbf{b}) \, p(\mathbf{y}_d | \boldsymbol{\pi}) \, p(\mathbf{z}_d | \boldsymbol{\psi}) \\ &= \prod_{i,j} \left[\pi_i \, \psi_j \, \sigma(\mathbf{u}_i^T \mathbf{v}_j + b_i + b_j) \right]^{y_{di} z_{dj}} \end{aligned}$$

while $p(o_{d'} = \mathsf{F} | \mathbf{y}_{d'}, \mathbf{z}_{d'}, \mathbf{U}, \mathbf{V}, \mathbf{b}) = \prod_{i,j} (1 - \sigma(\mathbf{u}_i^T \mathbf{v}_j + b_i + b_j))^{y_{d'i} z_{d'j}}$ is the odds of censoring pair d' if its indexes were known. The censored indexes $\mathbf{y}_{d'}$ and $\mathbf{z}_{d'}$ are unknown; by including their prior and marginalizing over them, $p(o_{d'} = \mathsf{F} | \boldsymbol{\vartheta})$ is a mixture of IJ components.

The joint density of $\mathscr{D}$ and the unobserved variables $\boldsymbol{\theta} \doteq \{\boldsymbol{\vartheta}, \{\mathbf{y}_{d'}, \mathbf{z}_{d'}\}_{d'=1}^{D'}\}$ depends on further priors on $\boldsymbol{\vartheta}$, for which we choose Dirichlet priors for $p(\boldsymbol{\pi}) = \mathcal{D}(\boldsymbol{\pi}; \boldsymbol{\alpha}_0)$ and $p(\boldsymbol{\psi}) = \mathcal{D}(\boldsymbol{\psi}; \boldsymbol{\alpha}_0)$. The other priors are fully factorized Gaussians, with $p(\mathbf{U}) = \prod_i \mathcal{N}(\mathbf{u}_i; \mathbf{0}, \tau_u^{-1} \mathbf{I})$ and $p(\mathbf{V}) = \prod_j \mathcal{N}(\mathbf{v}_j; \mathbf{0}, \tau_v^{-1} \mathbf{I})$ and, with some overloaded notation, $p(\mathbf{b}) = \prod_i \mathcal{N}(b_i; 0, \tau_b^{-1}) \prod_j \mathcal{N}(b_j; 0, \tau_b^{-1})$.

The joint density in Fig. 1 decomposes as

$$\begin{aligned} p(\mathscr{D}, \boldsymbol{\theta}) &= p(\mathscr{D} | \{\mathbf{y}_{d'}, \mathbf{z}_{d'}\}, \boldsymbol{\vartheta}) \, p(\{\mathbf{y}_{d'}, \mathbf{z}_{d'}\} | \boldsymbol{\pi}, \boldsymbol{\psi}) \, p(\boldsymbol{\vartheta}) \\ &= \prod_{i,j} \sigma(\mathbf{u}_i^T \mathbf{v}_j + b_i + b_j)^{c_{ij}} [1 - \sigma(\mathbf{u}_i^T \mathbf{v}_j + b_i + b_j)]^{\sum_{d'} y_{d'i} z_{d'j}} \\ &\quad \cdot \prod_i \pi_i^{c_i + \sum_{d'} y_{d'i}} \cdot \prod_j \psi_j^{c_j + \sum_{d'} z_{d'j}} \cdot p(\mathbf{U}) \, p(\mathbf{V}) \, p(\mathbf{b}) \, p(\boldsymbol{\pi}) \, p(\boldsymbol{\psi}) \, , \end{aligned} \tag{3}$$

where the uncensored data likelihood was regrouped using observation counts $c_{ij} \doteq \sum_d y_{di} z_{dj} \in \{0, 1, 2, \ldots, D\}$ for each pair (i, j), and marginal counts $c_i \doteq \sum_d y_{di}$ and $c_j \doteq \sum_d z_{dj}$. Note that $\sum_{i,j} c_{ij} = D$. Marginalizing $p(\mathscr{D}, \boldsymbol{\theta})$ over $\{\mathbf{y}_{d'}, \mathbf{z}_{d'}\}$ gives a mixture of $\binom{D'+IJ-1}{IJ-1}$ components, each representing a different way of assigning D' indistinguishable F's to IJ distinguishable bins, or assigning nonnegative counts c'_{ij} with $\sum_{i,j} c'_{ij} = D'$ to a "negative class count matrix".

At first glance of (3), it would seem as if inference would still scale with IJ, and that we have done nothing more than match the bilinear softmax's $\mathcal{O}(IJ)$ computational burden through the introduction of D'. The following section is devoted to developing a variational approximation, and with it a practically scalable inference scheme that relies on various "negative background" caches.

3 Variational Bayes

To find a scalable yet Bayesian inference procedure, we approximate $p(\boldsymbol{\theta} | \mathscr{D})$ with a factorized surrogate density $q(\boldsymbol{\theta})$, found by maximizing a variational lower bound to $\log p(\mathscr{D})$ [13]. First, we lower-bound each logistic function in (3) by associating a parameter ξ_{ij} with it [5]. Dropping subscripts, each bound would be $\sigma(\pm a) \geq \sigma(\xi) \exp(-\lambda(\xi) \left(a^2 - \xi^2\right) \pm \frac{a}{2} - \frac{\xi}{2})$, where the lower bound on $1 - \sigma(a)$ is that of $\sigma(-a)$ above. The bound depends on the deterministic function $\lambda(\xi) \doteq \frac{1}{2\xi}[\sigma(\xi) - \frac{1}{2}]$. Let $\boldsymbol{\xi} \doteq \{\xi_{ij}\}$ denote the set of logistic variational parameters, and substitute the bound into (3) to get $p(\mathscr{D}, \boldsymbol{\theta}) \geq p_{\boldsymbol{\xi}}(\mathscr{D}, \boldsymbol{\theta})$. Our variational objective $\mathcal{L}_{\boldsymbol{\xi}}[q]$, as a function of $\boldsymbol{\xi}$ and functional of q, follows from

$$\log p(\mathscr{D}) \geq \log \int p_{\boldsymbol{\xi}}(\mathscr{D}, \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} \geq \int q(\boldsymbol{\theta}) \log \frac{p_{\boldsymbol{\xi}}(\mathscr{D}, \boldsymbol{\theta})}{q(\boldsymbol{\theta})} \, \mathrm{d}\boldsymbol{\theta} \doteq \mathcal{L}_{\boldsymbol{\xi}}[q] \, , \tag{4}$$

which will be maximized with respect to q and $\boldsymbol{\xi}$. Our choice of factorization of q is

$$q(\boldsymbol{\theta}) \doteq \prod_i q(b_i) \prod_k q(u_{ik}) \cdot \prod_j q(b_j) \prod_k q(v_{jk}) \cdot \prod_{d'} q(\mathbf{y}_{d'}) \, q(\mathbf{z}_{d'}) \cdot q(\boldsymbol{\pi}) \, q(\boldsymbol{\psi}) \, . \tag{5}$$

The factors approximating each symbol's features in $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{b}$ are chosen to be a Gaussian, for example $q(u_{ik}) = \mathcal{N}(u_{ik}; \mu_{ik}, \omega_{ik}^{-1})$. The approximating factors $q(\boldsymbol{\pi})$ and $q(\boldsymbol{\psi})$ are both Dirichlet.

For the purpose of obtaining a scalable algorithm, the most important parameterizations are for the categorical (discrete) factors $q(\mathbf{y}_{d'})$ and $q(\mathbf{z}_{d'})$. As D' is potentially large, the parameters of $q(\mathbf{y}_{d'})$ will be tied. *This tying of parameters is the key to achieving a scalable algorithm.* We let all $q(\mathbf{y}_{d'})$ share the same parameter vector $\mathbf{s}$ on the probability simplex, such that $q(\mathbf{y}_{d'}) = \prod_i s_i^{y_{d'i}}$ for all d'. Similarly, all $q(\mathbf{z}_{d'})$ share probability vector $\mathbf{t}$, such that $q(\mathbf{z}_{d'}) = \prod_j t_j^{z_{d'j}}$ for all d'.

3.1 Scalable inference

Let graph $\mathcal{G} = \{(i,j) : c_{ij} > 0\}$ be the sparse set of all observed pair indexes. As there are IJ logistic variational parameters ξ_{ij}, we shall divide them into two sets, those with indexes in $\mathcal{G}$, and those without. Therefore ξ_{ij} shall be optimized for when $(i,j) \in \mathcal{G}$, while the ξ_{ij}'s shall *share the same parameter value* ξ^* for $(i,j) \notin \mathcal{G}$. Even though the form of (3) suggests that we would need two versions of ξ_{ij}, one for the bounded σ-term, and one its opposite, this is not required, as the optimization of the bound is symmetric. When ξ_{ij} maximizes $\mathcal{L}$ on the bounded σ-term, it simultaneously maximizes $\mathcal{L}$ on the bounded $(1-\sigma)$-term. We'll use the shorthand $\lambda_{ij} \doteq \lambda(\xi_{ij})$ for $(i,j) \in \mathcal{G}$; similarly, λ^* denotes $\lambda(\xi^*)$ when $(i,j) \notin \mathcal{G}$.

Below, we show how an update of $\prod_k q(u_{ik})$ only requires a sparse sum over $j \in \mathcal{G}(i)$, and not over all indexes j. This tractable update is a result of using

1. $\xi_{ij} = \xi^*$ (and hence $\lambda_{ij} = \lambda^*$) for all $j \notin \mathcal{G}(i)$;
2. $c_{ij} = 0$ for all $j \notin \mathcal{G}(i)$;
3. $\mathbb{E}_q[y_{d'i}] = s_i$ for all $d' = 1, \ldots, D'$;
4. $\mathbb{E}_q[z_{d'j}] = t_j$ for all $d' = 1, \ldots, D'$.

3.2 Gaussian updates for $q(u_{ik})$

As an example of a scalable update, we present here a bulk update of $\prod_k q(u_{ik})$. The bulk update is faster than sequentially maximizing $\mathcal{L}_{\boldsymbol{\xi}}[q]$ with respect to each of them in turn. For clarity, we assume that $b_i = b_j = 0$ are clamped at zero.

We first solve for the maximum of $\mathcal{L}$ with respect to a full Gaussian (not factorized) approximation $\tilde{q}(\mathbf{u}_i) = \mathcal{N}(\mathbf{u}_i; \boldsymbol{\mu}_i, \mathbf{P}_i^{-1})$. The fully factorized $q(u_{ik})$ can then be recovered from the intermediate approximation $\tilde{q}(\mathbf{u}_i)$ as those that minimize the Kullback-Leibler divergence $\mathrm{D}_{\mathrm{KL}}(\prod_k q(u_{ik}) \| \tilde{q}(\mathbf{u}_i))$: this is achieved when the means of $q(u_{ik})$ match that of $\tilde{q}(\mathbf{u}_i)$, while their precisions match the diagonal precision of $\tilde{q}(\mathbf{u}_i)$. The updates rely on careful caching, which we'll first illustrate through $\tilde{q}$'s precision matrix. $\mathcal{L}$ is maximized when $\tilde{q}(\mathbf{u}_i)$ has as natural parameters a precision matrix

$$\mathbf{P}_i = \sum_{j \in \mathcal{G}(i)} c_{ij} \cdot 2\lambda_{ij} \cdot \mathbb{E}_q\big[\mathbf{v}_j \mathbf{v}_j^T\big] + \overbrace{\sum_{d'} \sum_j \mathbb{E}_q[y_{d'i} \, z_{d'j}] \cdot 2\lambda_{ij} \cdot \mathbb{E}_q\big[\mathbf{v}_j \mathbf{v}_j^T\big]}^{\text{(a)}} + \tau_u \mathbf{I} \tag{6}$$

and a mean-times-precision vector $\mathbf{m}_i$, which we will state later. Looking at $\mathbf{P}_i$ in (6), a very undesirable sum over all d' and j is required in (a). We endeavoured that the update would be *sparse*, and only sum over observed indexes in $\mathcal{G}(i) \doteq \{j : (i,j) \in \mathcal{G}\}$. The benefit of the shared variational parameters now becomes apparent. With $\mathbb{E}_q[y_{d'i}\, z_{d'j}] = s_i t_j$ and $\lambda_{ij} = \lambda^*$ when $(i,j) \notin \mathcal{G}$, the sum in (a) decomposes as

$$(\mathrm{a}) = \sum_{j \in \mathcal{G}(i)} s_i t_j D' \cdot 2(\lambda_{ij} - \lambda^*)\, \mathbb{E}_q\big[\mathbf{v}_j \mathbf{v}_j^T\big] + s_i D' \cdot 2\lambda^* \cdot \overbrace{\sum_j t_j \mathbb{E}_q\big[\mathbf{v}_j \mathbf{v}_j^T\big]}^{\text{negative background } \mathbf{P}_\ominus} \; .$$

Barring the "negative background" term, only a sparse sum that involves observed pairs is required. This background term is rolled up into a global *item*-background cache, which is computed once before updating all $q(u_{ik})$. Throughout the paper, the $\ominus$ symbol will denote an item-background cache. The cache $\mathbf{P}_\ominus \doteq \sum_j t_j\, \mathbb{E}_q[\mathbf{v}_j \mathbf{v}_j^T]$ is used in each precision matrix update, for example

$$\mathbf{P}_i = s_i D' \cdot 2\lambda^* \cdot \mathbf{P}_\ominus + \sum_{j \in \mathcal{G}(i)} \Big(c_{ij} \cdot 2\lambda_{ij} + s_i t_j D' \cdot 2(\lambda_{ij} - \lambda^*)\Big) \mathbb{E}_q\big[\mathbf{v}_j \mathbf{v}_j^T\big] + \tau_u \mathbf{I} \, .$$

We've deliberately laboured the above decomposition of an expensive update into a background cache and a sparse sum over actual observations, as it serves as a template for other parameter updates. Turning to the mean-times-precision vector $\mathbf{m}_i \doteq \mathbf{P}_i \boldsymbol{\mu}_i$ of $\tilde{q}(\mathbf{u}_i)$, we find that

$$\mathbf{m}_i = \mathbb{E}_q \left[\frac{1}{2} \sum_{j \in \mathcal{G}(i)} c_{ij} \mathbf{v}_j - \frac{1}{2} \sum_{d'} \sum_j y_{d'i}\, z_{d'j} \mathbf{v}_j \right] . \qquad (7)$$

To find $\mathbf{m}_i$, an additional cache is added to the item-background cache, and are computed once before any $q(u_{ik})$ updates: $\mathbf{m}_\ominus \doteq \sum_j t_j \mathbb{E}_q[\mathbf{v}_j]$. The final mean-times-precision update is

$$\mathbf{m}_i = \frac{1}{2} \left(\sum_{j \in \mathcal{G}(i)} c_{ij} \mathbb{E}_q\big[\mathbf{v}_j\big] - s_i D' \mathbf{m}_\ominus \right) , \qquad (8)$$

and again only sums over $j \in \mathcal{G}(i)$ and not all $j = 1, \ldots, J$. There are of course additional variational parameters ξ_{ij}, and they are computed and discarded when needed.

3.3 Bilinear softmax gradients

The connection between this model and a bilinear softmax model can be seen when the biases are ignored. Consider the gradient of $\mathcal{L}$ with respect to *mean* parameter $\boldsymbol{\mu}_i$,

$$\nabla \mathcal{L}(\boldsymbol{\mu}_i) = -\mathbf{P}_i \boldsymbol{\mu}_i + \frac{1}{2} \left(\sum_{j \in \mathcal{G}(i)} c_{ij} \mathbb{E}_q\big[\mathbf{v}_j\big] - D' \sum_j s_i t_j \mathbb{E}_q\big[\mathbf{v}_j\big] \right) . \qquad (9)$$

The gradient $\nabla \mathcal{L}(\boldsymbol{\mu}_i)$ is zero at (7), which was stated, together with (6), in terms of *natural* parameters. As $\mathcal{L}(\boldsymbol{\mu}_i)$ is quadratic, it can be exactly maximized; furthermore, the maximum with respect to $\mathbf{P}_i$ is attained at the negative Hessian $\mathbf{P}_i = -\nabla^2 \mathcal{L}(\boldsymbol{\mu}_i)$, given in (6). The curvature of the bound, as a function of $\boldsymbol{\mu}_i$, directly translates into our posterior approximation's uncertainty of $\mathbf{u}_i$. The log likelihood of a softmax model would be $L = \sum_d \log p(i_d, j_d)$, with the likelihood of each pair defined by (1). The gradient of the log likelihood is therefore

$$\nabla L(\mathbf{u}_i) = \sum_{j \in \mathcal{G}(i)} c_{ij} \mathbf{v}_j - D \sum_j w_{ij} \mathbf{v}_j \, , \qquad (10)$$

with weights $w_{ij} \doteq \mathrm{e}^{\mathbf{u}_i^T \mathbf{v}_j} / \sum_{i',j'} \mathrm{e}^{\mathbf{u}_{i'}^T \mathbf{v}_{j'}}$ that sum to one over all IJ options. The weights in (9) were simply $s_i t_j$, and also sum to one over all options. The difference between (9) and (10) is that $s_i t_j$ is used as a *factorized substitute* for w_{ij}. This simplification allows the convenience that none of the updates described in Sec. 3.1 need to be stochastic, and substitute functions, as employed by noise contrastive estimation to maximize L, are not required. (The Hessian $\nabla^2 L(\mathbf{u}_i)$ contains a *double-sum* over indexes j.) Considering the two equations above, one might expect to set hyperparameter D' to $D' = D$, and in Sec. 4 we show that this is a reasonable choice.

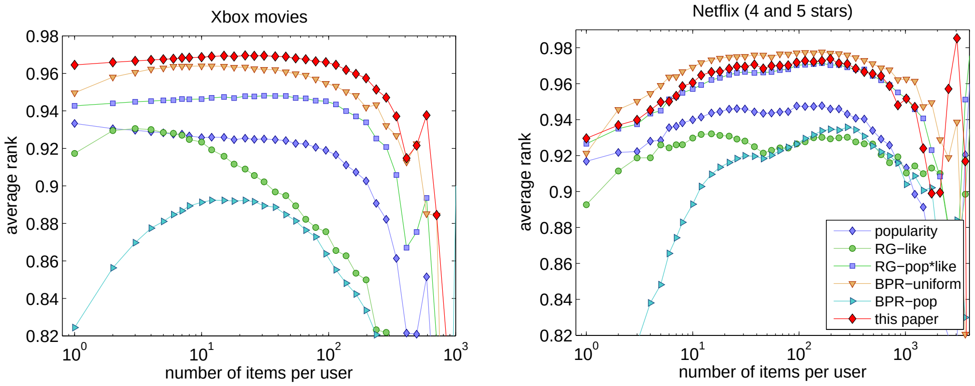

Figure 2: The rank $R(i, j^\star)$ in (12), averaged over users and grouped logarithmically by c_i. The *top* evaluation is on the *Xbox movies* sample, while the *bottom* one is on the "implicit feedback" *Netflix (4 and 5 stars)* set.

4 Evaluation

A key application for modelling paired $(\textit{user}, \textit{item})$ symbols is large-scale recommendation systems and we evaluate predictions from the model on two large data sets. The *Xbox movies* data is a sample of 5.6×10^7 views for 6.2×10^6 users on a sub-catalogue of around 1.2×10^4 movies [10]. To evaluate on data known in the Machine Learning community, the four- and five-starred ratings from the Netflix prize data set were used to simulate a stream of "implicit feedback" $(\textit{user}, \textit{item})$ pairs in the *Netflix (4 and 5 stars)* data. We refer the reader to [10] for a complete data set description. Let's first consider how predictions are made from the model.

4.1 Making predictions

Our original desideratum was to infer the probability of symbol j, conditional on the other symbol being i, and the observed data. Bayesian marginalization requires us to average the predictions over the model parameter posterior distribution. Here it is an analytically intractable task, which we approximate by using q as a surrogate for the true posterior. Firstly,

$$p(o = \mathsf{T}|\mathbf{y}, \mathbf{z}, \mathscr{D}) \approx \int p(o = \mathsf{T}|\mathbf{y}, \mathbf{z}, \boldsymbol{\vartheta}) q(\boldsymbol{\vartheta})\, \mathrm{d}\boldsymbol{\vartheta} = \int \sigma(a_{ij})\, \mathcal{N}(a_{ij}; \mu_{ij}, \sigma_{ij}^2)\, \mathrm{d}a_{ij} \approx \sigma(x_{ij})$$

if $y_i = z_j = 1$. The random variable a_{ij} was defined as $a_{ij} \doteq \mathbf{u}_i^T \mathbf{v}_j + b_i + b_j$, with its density approximated with its first two moments under q, i.e. $\mu_{ij} \doteq \mathbb{E}_q[a_{ij}]$ and $\sigma_{ij}^2 \doteq \mathbb{E}_q[(a_{ij} - \mu_{ij})^2]$. With $x_{ij} \doteq \mu_{ij} \,/ (1 + \pi \sigma_{ij}^2 / 8)^{1/2}$, the final approximation of a logistic Gaussian integral follows from [6]. Again using q, the posterior density of symbol j, provided that the first symbol is i, is approximately proportional to (writing "T" for "$o = \mathsf{T}$" for brevity)

$$p(z_j = 1|\mathsf{T}, y_i = 1, \mathscr{D}) \underset{\approx}{\propto} p(\mathsf{T}|y_i = z_j = 1, \mathscr{D}) \int p(z_j = 1|\boldsymbol{\psi}) q(\boldsymbol{\psi})\, \mathrm{d}\boldsymbol{\psi} = \sigma(x_{ij})\, \mathbb{E}_q[\psi_j] \,. \quad (11)$$

Hence $p(z_j = 1|o = \mathsf{T}, y_i = 1, \mathscr{D}) \approx \sigma(x_{ij})\, \mathbb{E}_q[\psi_j] \Big/ \sum_{j'} \mathbb{E}_q[\psi_{j'}]\, \sigma(x_{ij'})$, normalizing to one.

4.2 Ranking items

For each user, one item was randomly removed to create a test set. To mimic a real scenario in the simplest possible way, each user's non-viewed items were ranked, and the position of the test item noted. We are interested in the rank of held out item $j^\star$ for user i on a $[0, 1]$ scale,

$$R(i, j^\star) \doteq \frac{1}{J - |\mathcal{G}(i)|} \sum_{j \notin \mathcal{G}(i)} \mathbb{I}\Big[f_{ij^\star} > f_{ij} \Big] \,, \quad (12)$$

where f_{ij} indicates the score given by (11) or any alternative algorithm. (In a real system $q(\boldsymbol{\theta})$ would be employed in various utility functions that cater for diversity, freshness, exposure of tail items, etc.)

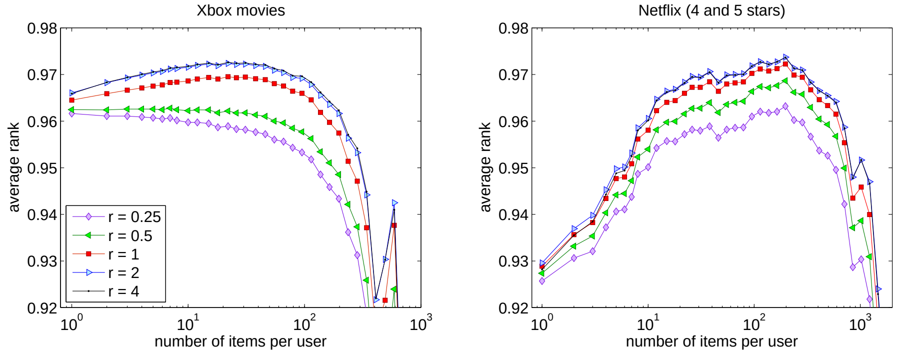

Figure 3: The average rank $R(i, j^\star)$ in (12), grouped logarithmically by c_i, for varying values of r in $D' = rD$.

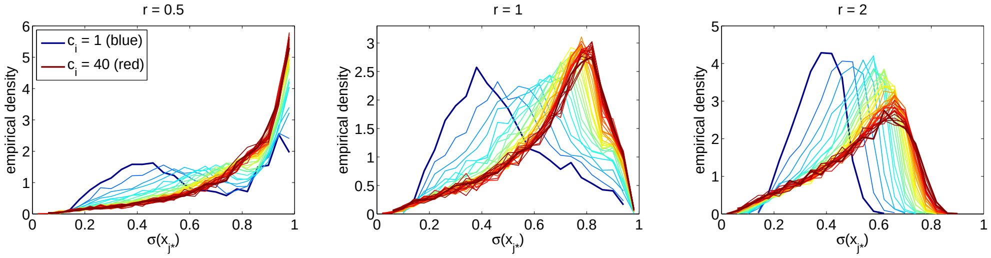

Figure 4: The empirical densities of $\sigma(x_{j^\star})$, as defined in (11), over all held-out items $j^\star$ in the *Netflix (4 and 5 stars)* set. The densities are sliced according to $c_i = 1, \ldots, 40$ for different values of r in $D' = rD$.

In Fig. 2, we facet the average rank by c_i, the number of movie views per user. As the evaluation is over 6 million users, this gives a more representative perspective than reporting a single average. Apart from ranking by *popularity* c_j, which would be akin to only factorizing with $s_i t_j$, we compare against two other baselines. *BPR-uniform* and *BPR-pop* represent different versions of the Bayesian Personalized Ranking algorithm [12], which optimizes a rank metric directly against either the data distribution of items (*BPR-uniform*, with missing items are sampled uniformly during stochastic optimization), or a tilted distribution aimed at personalizing recommendations regardless of an item's popularity (*BPR-pop*, with missing items sampled proportional to their popularity). Their hyperparameters were set using cross-validation. For the Random Graph model [10], rankings are shown with pure personalization (*RG-like*) and with an item popularity rate factored in (*RG-pop*like*). The comparison in Fig. 2 is drawn using $K = 20$ dimensions, $D' = D$ and hyperparameters set to one. For *Xbox movies*, the model outperforms all alternatives that we compared against. *BPR-uniform*, optimizing (12) directly, performs slightly better on the less sparse Netflix set (the Xbox usage sample is much sparser, as it is easier to rate many movies than to view as many).

We surmised in Section 3.1 that $D' = D$ is a reasonable hyperparameter setting, and Fig. 3 validates this claim. The figure shows the average held-out rank on the *Netflix (4 and 5)* set for various settings of D' through $D' = rD$. The average rank improves beyond $r = 1$, but empirically slowly decreases beyond $r = 2$. To provide insight into the *"censoring"* step, Fig. 4 accompanies Fig. 3, and shows the empirical density of the Bernoulli variable $\sigma(x_{j^\star})$ for held-out items $j^\star$. We break the empirical density down over users that appear in $c_i = 1, 2, 3, \ldots, 40$ pairs. Given that the held-out pairs were observed, the Bernoulli variable should be *true*, and the density of $\sigma(x_{j^\star})$ shifts right as c_i becomes bigger. The effect of having to explain less ($r = \frac{1}{2}$) or more ($r = 2$) censored pairs is also visible in the figure. There is also a slight benefit in increasing K. The average rank $\hat{R}_{20}$ for $K = 20$ is 0.9649, using $r = 1$. An increased latent dimensionality gives $\hat{R}_{30} - \hat{R}_{20} = 1.07 \times 10^{-4}$, $\hat{R}_{40} - \hat{R}_{20} = 1.73 \times 10^{-4}$, and $\hat{R}_{50} - \hat{R}_{20} = 0.87 \times 10^{-4}$.

Finally, careful caching and parallelization one could obtain a fast implementation. For *Xbox movies*, updating all item-related parameters took 69 seconds on a 24-core (Intel Xeon 2.93Ghz) machine, and updating all user-related parameters for 6 million users took 83 seconds.

5 Summary and outlook

We presented a novel model for pairs of symbols, and showed state of the art results on a large-scale movies recommendation task. Scalability was achieved by factorizing the popularity or selection step via $\pi_i \psi_j$, and employing "site-independent" variational bounds through careful parameter tying. This approach might be too simplistic; an extension would be to use a N-component mixture model to select pairs with odds $\sum_{n=1}^{N} \pi_{in} \psi_{jn}$, and perform inference with Gibbs sampling.

It is worth noting that Böhning [2] and Bouchard [3] provide lower bounds to the logarithm of (1). We originally embarked on a variational approximation to a posterior with (1) as likelihood using Bouchard's bound, for which bookkeeping like Sec. 3.1's was done. However, with realistically large I and J, solutions were trivial, as the means of the variational posterior approximations for $\mathbf{u}_i$ and $\mathbf{v}_j$ were zero. We leave Böhning's bound to future work.

References

[1] Y. Bengio and J.-S. Senécal. Quick training of probabilistic neural nets by importance sampling. In *Artificial Intelligence and Statistics*, 2003.

[2] D. Böhning. Multinomial logistic regression algorithm. *Annals of the Institute of Statistical Mathematics*, 44:197–200, 1992.

[3] G. Bouchard. Efficient bounds for the softmax and applications to approximate inference in hybrid models. In *NIPS 2007 Workshop on Approximate Inference in Hybrid Models*, 2007.

[4] M. U. Gutmann and A. Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13:307–361, 2012.

[5] T. Jaakkola and M. Jordan. A variational approach to Bayesian logistic regression problems and their extensions. In *Artificial Intelligence and Statistics*, 1996.

[6] D. J. C. MacKay. The evidence framework applied to classification networks. *Neural Computation*, 4(5):698–714, 1992.

[7] T. Mikolov, I. Sutskever, K. Cheni, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119. 2013.

[8] A. Mnih and Y. W. Teh. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1751–1758, 2012.

[9] A. Mnih and Y. W. Teh. Learning label trees for probabilistic modelling of implicit feedback. In *Advances in Neural Information Processing Systems 25*, pages 2825–2833. 2012.

[10] U. Paquet and N. Koenigstein. One-class collaborative filtering with random graphs. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 999–1008, 2013.

[11] U. Paquet, N. Koenigstein, and O. Winther. Scalable Bayesian modelling of paired symbols. *arXiv:1409.2824*, 2014.

[12] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Uncertainty in Artificial Intelligence*, pages 452–461, 2009.

[13] S. R. Waterhouse, D.J.C. MacKay, and A. J. Robinson. Bayesian methods for mixtures of experts. In *Advances in Neural Information Processing Systems 8*, pages 351–357. 1996.