Modeling Item Selection and Relevance for Accurate Recommendations: a Bayesian Approach

Nicola Barbieri
University Of Calabria,
ICAR-CNR
via Bucci 41c
Rende (CS), Italy
nbarbieri@deis.unical.it

Gianni Costa
ICAR-CNR
via Bucci 41c
Rende (CS), Italy
costa@icar.cnr.it

Giuseppe Manco
ICAR-CNR
via Bucci 41c
Rende (CS), Italy
manco@icar.cnr.it

Riccardo Ortale
ICAR-CNR
via Bucci 41c
Rende (CS), Italy
ortale@icar.cnr.it

ABSTRACT

We propose a bayesian probabilistic model for explicit preference data. The model introduces a generative process, which takes into account both item selection and rating emission to gather into communities those users who experience the same items and tend to adopt the same rating pattern. Each user is modeled as a random mixture of topics, where each topic is characterized by a distribution modeling the popularity of items within the respective user-community and by a distribution over preference values for those items. The proposed model can be associated with a novel *item-relevance* ranking criterion, which is based both on item popularity and user's preferences. We show that the proposed model, equipped with the new ranking criterion, outperforms state-of-art approaches in terms of accuracy of the recommendation list provided to users on standard benchmark datasets.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retriveal – Information Filtering; H.2.8.d [**Information Technology and Systems**]: Database Applications - Data Mining; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Experimentation, Performance

1. INTRODUCTION

With the increasing volume of information, products, services (or, more generally, items) available on the Web, the role of *Recommender Systems (RS)* and the importance of highly-accurate recommendation techniques have become a major concern both in e-commerce and academic research. In particular, the goal of a RS is to provide users with not trivial recommendations, that are useful to directly experience potentially interesting items. State-of-the art recommendation methods have been largely approached from a Collaborative Filtering (CF) perspective, which essentially consists in the posterior analysis of past interactions between users and items, aimed to identify suitable preference patterns in users' preference data.

Traditional CF approaches have focused on the minimization of the prediction error, by employing standard statistical error metrics such as the popular *Root Mean Squared Error (RMSE)*. The underlying assumption was that minimum improvements in terms of RMSE would have determined an increase of recommendation accuracy, which is the actual goal. However, a recent study [7] has empirically shown that a monotonic relation between prediction and recommendation accuracy does not exist. In fact, the main result of [7] is that Pure-SVD, which is not designed to minimize prediction error, achieves the best result in terms of the accuracy of the recommendation list provided to the users. By following this observation, it has been subsequently shown in [3] that probabilistic approaches based on latent-factor models achieve the highest recommendation accuracy (and, as matter of fact, outperform Pure-SVD), though being designed to maximize the likelihood of the underlying model rather than attempting to minimize the prediction error.

Following the line drawn in [7, 3], in this paper we propose a new *Bayesian User Community Model (UCM)* and comparatively investigate its recommendation accuracy. Bayesian UCM relies on a generative process, which takes into account both item selection and rating emission, so that those users who experience the same items and tend to adopt the same rating pattern are gathered into communities. The individual users are modeled as a random mixture of topics, where the individual topic is characterized by both a distribution modeling the popularity of items within the respective user-community and by a distribution over item ratings.

RecSys'11, October 23–27, 2011, Chicago, Illinois, USA.

A key difference with respect to conventional probabilistic approaches to recommendation is that Bayesian UCM allows *free-prediction* [10]. While most of the conventional probabilistic techniques focus on *forced-prediction*, which explicitly requires to predict the preference value for each observed user-item pair, the goal of Bayesian UCM is to model item selection and rating prediction simultaneously. A novel item-ranking scheme, referred to as *item selection and relevance ranking*, is used for this purpose. Such a scheme accommodates both the components for item popularity within the individual user communities and rating prediction.

Bayesian UCM reinterprets the former UCM [4] through a Bayesian modeling approach, that is better suited to the sparseness of the preference data and less susceptible to overfitting. Additionally, Bayesian UCM allows a simpler and more elegant procedure for the estimation of model parameters through Gibbs sampling [5]. The experimental evaluation in Sec. 4 shows that the proposed approach outperforms state-of-art competitors in recommendation accuracy.

The outline of this paper is as follows. Section 2 introduces some preliminaries. Section 3 develops the Bayesian UCM model. Section 4 presents the results on an intensive comparative evaluation of the Bayesian UCM model. Finally, Section 5 concludes and highlights further research.

2. PRELIMINARIES

We introduce in this section the notation used throughout the paper along with some preliminary concepts.

Let $\mathcal{U} = \{u_1, \ldots, u_M\}$ be a set of M users and $\mathcal{I} = \{i_1, \ldots, i_N\}$ a set of N items. User' s preferences can be represented as a $M \times N$ matrix $\mathbf{R}$, whose generic entry r_i^u denotes the rating value (i.e., the degree of preference) assigned by user u to item i. For each pair $\langle u, i \rangle$, rating values r_i^u fall within a limited integer range $\mathcal{V} = \{0, \ldots, V\}$, where 0 represents an unknown rating and V is the maximum degree of preference. Notation $\overline{r}_{\mathbf{R}}$ denotes the average rating among all those ratings $r_i^u > 0$.

The number of users M as well as the number of items N are very large (typically with $M >> N$) and, in practical applications, the rating matrix $\mathbf{R}$ is characterized by an an exceptional sparseness (e.g., more than 95%), since the individual users tend to rate a limited number of items. The set of items rated by user u is denoted by $\mathcal{I}_{\mathbf{R}}(u) = \{i \in \mathcal{I} | r_i^u > 0\}$. Dually, $\mathcal{U}_{\mathbf{R}}(i) = \{u \in \mathcal{U} | r_i^u > 0\}$ is the set of all those users, who rated item i. Any user u with a rating history, i.e., such that $\mathcal{I}_{\mathbf{R}}(u) \neq \emptyset$ is said to be an active user.

Given an active user u, the goal of a RS is to provide u with a recommendation list $\mathcal{R} \subseteq \mathcal{I}$ including unexperienced items (i.e., $\mathcal{R} \cap \mathcal{I}_{\mathbf{R}}(u) = \emptyset$), that are expected to be of interest to u. This clearly involves predicting the interest of u into unrated items. In this paper, we are interested in the adoption of probabilistic approaches to CF for the generation of recommendation lists, since these offer some relevant advantages (summarized in Section 3). In particular, we focus on probabilistic approaches based on latent factors, whose idea is that each preference observation $\langle u, i \rangle$ is generated by one of multiple possible states, which informally explains the reason why u rated i.

[]⟹Nuova sezione⟸[]

In the following, to keep notation uncluttered, we will write $P(r, u, i)$ to mean the joint probability $P(R = r, U = u, I = i)$, where R, U and I are random variables taking values r, u and i, respectively, from the set of rating values $\mathcal{V}$, the set of users $\mathcal{U}$ and the set of items $\mathcal{I}$. Likewise, the same notation will be also adopted for conditional probabilities (e.g., $P(r|u, i)$ corresponds to $P(R = r|U = u, I = i)$).

Based on the underlying mathematical model, probabilistic approaches allow the prediction of the expected interest of a user u into an item i in two different ways [10].

- *Forced prediction*: the probabilistic model provides an estimate of $P(r|u, i)$, which represents the conditional probability that user u assigns a rating value r given the item i;
- *Free prediction*: the item selection process is included in the probabilistic model, which is typically based on the estimate of $P(r, i|u)$. In this case we are interested in predicting both the item selection and the preference of the user for each selected item. $P(r, i|u)$ can be factorized as $P(r|i, u)P(i|u)$; the resulting model still includes a component of forced prediction which however is weighted by the item selection component and thus allows a more precise estimate of user's preferences.

In general, a recommendation list $\mathcal{R}$ can be generated as follows:

- Let $\mathcal{C}$ be a set of d candidate recommendations to arbitrary items, not yet rated by u;
- Associate each item $i \in \mathcal{C}$ with a score p_i^u representing u's interest into i.
- Sort $\mathcal{C}$ in descending order of item scores p_i^u;
- Add the first k items from $\mathcal{C}$ to $\mathcal{R}$ and return the latter to user u.

A common framework in the evaluation of the predictive capabilities of a RS algorithm is to split the rating matrix $\mathbf{R}$ into two matrices $\mathbf{T}$ and $\mathbf{S}$, such that the former is used to train the RS, while the latter is used for validation purposes. By selecting a user u from $\mathbf{S}$, the set $\mathcal{C}$ of candidate recommendations is obtained by drawing upon $\mathcal{I} - \mathcal{I}_{\mathbf{T}}(u)$. The recommendation list $\mathcal{R}$ for u is then formed by following the foregoing generation process and the accuracy of $\mathcal{R}$ is ultimately assessed through a comparison with the items appearing in $\mathcal{I}_{\mathbf{S}}(u)$. Therein, the standard classification-based metrics, i.e., precision and recall, can be adopted to evaluate the recommendation accuracy of $\mathcal{R}$. Such metrics require the capability to distinguish between relevant and irrelevant recommendations.

Given a user u and a subset $\mathcal{T}_u^r \subseteq \mathcal{I}_{\mathbf{S}}(u)$ of relevant items, the degree of precision and recall of the k items within $\mathcal{R}$ is defined as shown next:

$$Rec(k) = \frac{1}{M} \sum_{u=1}^{M} \frac{|\mathcal{R} \cap \mathcal{T}_u^r|}{|\mathcal{T}_u^r|} \qquad Prec(k) = \frac{1}{M} \sum_{u=1}^{M} \frac{|\mathcal{R} \cap \mathcal{T}_u^r|}{k}$$

Item relevance can be measured in several different ways. Since explicit preference values are available, we consider as relevant all those items that received a rating greater than the average ratings in the training set, i.e.

$$\mathcal{T}_u^r = \{i \in \mathcal{I}_{\mathbf{S}}(u) | \mathbf{S}_i^u > \overline{r}_{\mathbf{T}}\}$$

The above definitions of precision and recall consider the amount of useful recommendations. A different perspective can be considered by assuming that a recommendation meets user satisfaction, if the user can find at least an interesting item in the recommendation list. This perspective is modeled by a different approach to measure the accuracy of the recommendation list, proposed in [7], that introduces a testing protocol, with which to accordingly tune the definition of precision and recall, that are henceforth referred to as user-satisfaction (US) precision and recall.

A different definition of relevant items is at the basis of the re-interpretation of standard precision and recall, namely:

$$\mathcal{T}_u^r = \{i \in \mathcal{I}_{\mathbf{S}}(u) | \mathbf{S}_i^u = V\}$$

Then, the following testing protocol can be applied to assess user satisfaction:

- For each user u and for each item $i \in \mathcal{T}_u^r$:
  - Generate the candidate list $\mathcal{C}$ by randomly drawing from $\mathcal{I}_{\mathbf{R}}(u) - (\mathcal{I}_{\mathbf{T}}(u) \cup \{i\})$.
  - Add i to $\mathcal{C}$.
  - Associate each item within $\mathcal{C}$ with a suitable score and sort $\mathcal{C}$ in descending order of item scores.
  - Consider the position of the item i in the ordered list: if i belongs to the top-k items, there is a *hit*; otherwise, there is a *miss*.

By definition, US recall for an interesting item can be either 0 (in the case of a failure) or 1 (in the case case of a hit). Likewise, US precision can be either 0 (in the case of a failure) or $\frac{1}{k}$ (in the case of a hit). The overall US precision and recall are defined in [7] as the below averages:

$$\begin{aligned} US - Recall(k) &= \frac{\#hits}{|\mathcal{T}_u^r|} \\ US - Precision(k) &= \frac{\#hits}{k \cdot |\mathcal{T}_u^r|} = \frac{US - Recall(k)}{k} \end{aligned}$$

A key role in the process of generating accurate recommendation lists is played by the schemes with which to rank items candidate for recommendation. [3] provides a comparative analysis of three possible such schemes, and studies their impact in the accuracy of the recommendation list. The results of such study can summarized as follows.

- Lower RMSE values do not necessarily imply improvements in recommendation accuracy. Cutting-edge probabilistic approaches, such as PMF [16], equipped with *expected-value* ($p_i^u = E[R|u,i]$) item-ranking schemes have been shown to perform poorly in terms of recommendation accuracy.

- Probabilistic CF methods were shown to outperform state-of-the-art competitors in terms of recommendation accuracy when equipped with the item selection scheme $p_i^u = P(i|u) = \sum_z P(z|u)P(i|z)$.

In the model proposed in this paper, we shall concentrate on a mix of *item selection and relevance ranking*. Formally, we aim at forcing the selection process to focus on relevant items, by counterbalancing the prediction probability with a component that represents the *predicted* relevance of an item i with respect to a given user u:

$$\begin{aligned} p_i^u &= P(i, r > \overline{r}_{\mathbf{T}}|u) \\ &= P(i|u)P(r > \overline{r}_{\mathbf{T}}|u, i) \\ &= \sum_z P(z|u)P(i|z)P(r > \overline{r}_{\mathbf{T}}|i, z) \end{aligned}$$

where $P(r > \overline{r}_{\mathbf{T}}|i,z) = \sum_{r > \overline{r}_{\mathbf{T}}} P(r|i,z)$.

As a concluding remark, it is worth to highlight that probabilistic CF methods based on free-prediction are better suited to support the item-selection-and-relevance-ranking scheme.

3. A USER COMMUNITY MODEL

Probabilistic approaches assume that each preference observation is randomly drawn from the joint distribution of the random variables which model users, items and preference values (if available). Typically, the random generation process follows a bag of words assumption and preference observations are assumed to be generated independently. A key difference between probabilistic and deterministic models relies in the inference phase: while the latter approaches try to minimize directly the error made by the model, probabilistic approaches do not focus on a particular error metric; parameters are determined by maximizing the likelihood of the data, typically by employing an expectation-maximization procedure [5]. Furthermore, background knowledge can be explicitly modeled through prior probabilities, thus allowing a direct control on overfitting within the inference procedure [9]. By modeling prior knowledge, probabilistic approaches implicitly also solve the need for regularization, which affects traditional gradient-descent based latent factors approaches. In addition, when explicit preference values are available, probabilistic models can be used to model a distribution over rating values, which allows to infer confidence intervals and to determine the confidence of the model in providing a recommendation.

In this section we develop Bayesian UCM, a new probabilistic approach for explicit preference data. The devised Bayesian UCM model reformulates the basic UCM model [4] through a Bayesian approach. With respect to UCM, that relies on maximum likelihood estimation with multinomial priors for model inference, the new Bayesian formulation is both better suited to the sparsity of the rating matrix and less susceptible to overfitting. Moreover, it allows the development of a simpler and more elegant procedure for approximated parameter estimation based on Gibbs sampling [5].

Bayesian UCM relies on a generative process, which takes into account both item selection and rating emission. Each user is modeled as a random mixture of topics, where the individual topic is then characterized both by a distribution modeling item-popularity within the considered user-community and by a distribution over preference values for those items.

The main difference between the proposed Bayesian UCM model and the state-of-art probabilistic approaches to CF is the former is a free-prediction model. In fact, while most of the models accord with a missing-value perspective and, hence, are focused on the prediction of a preference value r_i^u given the pair $\langle u, i \rangle$, the Bayesian UCM model tries to also infer the tendency of a user to experience some items over others independent of her/his rating values. The Bayesian UCM model assumes that this tendency is influenced by im-

plicit and hidden factors which characterize each user community. To elucidate, a user may be pushed to experience a certain item because she/he belongs to a community in which the category of that item occurs with an high probability, although this has no impact on the rating assigned to the foresaid item category. The probability of observing an item is independent from the rating assigned, given the state of the latent variables. Moreover, free-prediction models are focused on both the estimation of a rating behaviour and the popularity of an item within each user community. An item which has received high ratings and has been experienced few times by the users belonging to the considered community could not have better chances of being recommended with respect to a popular item within the same community, which has received only ratings around the average.

The generative process behind the Bayesian UCM can be summarized as follows:

1. For each user $u \in \mathcal{U}$ sample user community-mixture components $\vec{\vartheta}_u \sim Dir(\vec{\alpha})$
2. For each topic (or equivalently user community) $z = \{1, \cdots, K\}$,
   (a) Sample item selection components $\vec{\varphi}_z \sim Dir(\vec{\beta})$
   (b) Sample rating probabilities $\vec{\varepsilon}_z \sim Dir(\vec{\gamma})$
3. Sample the number of items for the user u, $N_u \propto Poisson(\mathcal{K})$
4. For $n = 1$ to N_u
   (a) Choose a user attitude $z_{u,n} \sim Discrete(\vec{\vartheta}_u)$
   (b) Choose an item $i_n \sim Multi(\varphi | z_{u,n})$
   (c) Generate a rating value for the chosen item according to the distribution $P(r|\vec{\varepsilon}_{z_{u,n},i_n})$

The corresponding graphical model is illustrated in Fig. 1

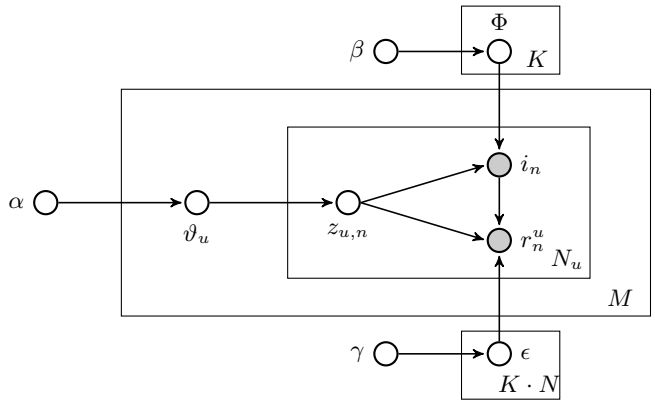


Figure 1: Bayesian User Community Model

3.1 Inference and Parameter Estimation

We here introduce inference and parameter estimation within the devised Bayesian UCM. The notation used in our discussion is summarized in Tab. 3.1. Given the hyperparameters $\vec{\alpha}$, $\vec{\beta}$ and $\vec{\gamma}$, the joint distribution of the data $\mathbf{R}$, the user-topic mixtures Θ, the item-selection components Φ, the rating probabilities Γ and the observation-topic assignments $\underline{Z}$, can be computed as:

$$P(\mathbf{R}, \underline{Z}, \Theta, \Phi, \Gamma | \vec{\alpha}, \vec{\beta}, \vec{\gamma}) = P(\mathbf{R}|\underline{Z}, \Phi, \Gamma) P(\underline{Z}|\Theta) \cdot P(\Theta|\vec{\alpha}) P(\Phi|\vec{\beta}) P(\Gamma|\vec{\gamma})$$

The complete data likelihood can be obtained by integrating over Θ, Φ and Γ:

$$P(\mathbf{R}, \underline{Z}|\vec{\alpha}, \vec{\beta}, \vec{\gamma}) = \int\int\int P(\mathbf{R}|\underline{Z}, \Phi, \Gamma) P(\underline{Z}|\Theta) \cdot P(\Theta|\vec{\alpha}) P(\Phi|\vec{\beta}) P(\Gamma|\vec{\gamma}) \mathrm{d}\Theta \mathrm{d}\Phi \mathrm{d}\Gamma$$

which, due to the conditional independence $\mathbf{R} \perp\!\!\!\perp \vec{\alpha} | \underline{Z}$, can be factored as:

$$P(\mathbf{R}, \underline{Z}|\vec{\alpha}, \vec{\beta}, \vec{\gamma}) = \int P(\underline{Z}|\Theta) P(\Theta|\vec{\alpha}) \mathrm{d}\Theta \int\int P(\mathbf{R}|\underline{Z}, \Phi, \Gamma) P(\Phi|\vec{\beta}) P(\Gamma|\vec{\gamma}) \mathrm{d}\Phi \mathrm{d}\Gamma$$

By rearranging the components and grouping the conjugate distributions, the complete data likelihood can be expressed as:

$$P(\mathbf{R}, \underline{Z}|\vec{\alpha}, \vec{\beta}, \vec{\gamma}) = \prod_{u=1}^{M} \frac{1}{\Delta(\vec{\alpha})} \int \prod_{k=1}^{K} \vartheta_{u,k}^{n_u^k + \alpha_k - 1} \mathrm{d}\vec{\vartheta}_u \prod_{i=1}^{N} \frac{1}{\Delta(\vec{\beta})\Delta(\vec{\gamma})} \int \prod_{k=1}^{K} \varphi_{i,k}^{n_i^k + \beta_i - 1} \mathrm{d}\vec{\varphi}_k \cdot \int \prod_{k=1}^{K} \prod_{r=1}^{V} \epsilon_{k,i,r}^{n_{i,r}^k + \gamma_r - 1} \mathrm{d}\vec{\varepsilon}_{k,i}$$

The latter is the starting point for the inference of all the topics underlying the generative process, as the conditioned distribution on $\underline{Z}$ can be written as:

$$P(\underline{Z}|\mathbf{R}, \vec{\alpha}, \vec{\beta}, \vec{\gamma}) = \frac{P(\underline{Z}, \mathbf{R}|\vec{\alpha}, \vec{\beta}, \vec{\gamma})}{P(\mathbf{R}|\vec{\alpha}, \vec{\beta}, \vec{\gamma})}$$

This formula is however intractable, mainly because the computation of the denominator requires a summation over an exponential number of terms. Gibbs Sampling [5] addresses this problem by defining a Markov chain, in which at each step inference can be accomplished by exploiting the full conditional $P(Z_n | \underline{Z}_{\neg n}, \mathbf{R}, \vec{\alpha}, \vec{\beta}, \vec{\gamma})$. In the latter, n denotes a single rating observation $n = \{u, i, r_i^u\}$, Z_n is the cell of the matrix $\underline{Z}$ which corresponds to this observation, and $\underline{Z}_{\neg n}$ denotes the remaining topic assignments. The chain is hence defined by iterating over all the available n states. The Gibbs Sampling algorithm estimates the probability of assigning the topic k to the n-th observation, given the assignment corresponding to all the other rating observations:

$$p(Z_n = k | \underline{Z}_{\neg n}, \mathbf{R}) \propto \frac{n_u^k + \alpha_k - 1}{\sum_{k'=1}^{K} (n_u^{k'} + \alpha_{k'}) - 1} \cdot \frac{n_i^k + \beta_i - 1}{\sum_{i'=1}^{N} \left(n_{i'}^k + \beta_{i'}\right) - 1} \cdot \frac{n_{i,r}^k + \gamma_r - 1}{\sum_{r'=1}^{V} (n_{r',i}^k + \gamma_{r'}) - 1} \quad (1)$$

SYMBOL	DESCRIPTION
M	#Users
N	# Items
$\mathbf{R}$	$M \times N$ Rating Matrix
K	# topics/user communities
$\vec{\alpha}$	K- vector, Dirichlet priors on user communities
$\vec{\beta}$	N-vector, Dirichlet priors on items
$\vec{\gamma}$	V-vector, Dirichlet priors on rating values
Θ	matrix of parameters $\vec{\vartheta}_u$
$\vec{\vartheta}_u$	mixing proportion of communities for the user u
z	topic variable
Φ	matrix of parameters $\vec{\varphi}_k$
$\vec{\varphi}_k$	mixing proportion of items for the community k
N_u	# preference observations for user u
Γ	matrix of parameters $\vec{\epsilon}_k$
$\vec{\epsilon}_k$	vector of rating distributions $\vec{\epsilon}_{k,i}$ for topic k
$\vec{\epsilon}_{k,i}$	distribution over rating values for the item i and the community k
n_i^k	# times that the item i has been assigned to topic k
$n_{i,r}^k$	# times that the rating r has been assigned to the item i when the topic is k
n_u^k	# times an item evaluated by u has been assigned to topic k
z_n	topic assignment for the observation $n = \langle u, i \rangle$
$z_{\neg n}$	topic assignment for all other observations except the current observation $n = \langle u, i \rangle$

Table 1: Summary of notation

Given the state of the markov chain, denoted my $\mathcal{M} = \mathbf{R}, \underline{Z}$, where $\underline{Z}$ encodes the topic assignment for each pair $\langle u, i \rangle \in \mathbf{R}$, we can obtain the multinomial parameters Φ and Θ and Γ noticing that, by algebraic manipulations, they reduce to Dirichlet distributions and can hence been estimated as the underlying expectations [8]:

$$\vartheta_{u,k} = \frac{n_u^k + \alpha_k}{N_u + \sum_{k=1}^{K} \alpha_k} \tag{2}$$

$$\varphi_{i,k} = \frac{n_i^k + \beta_i}{\sum_{i=1}^{N} n_i^k + \beta_i} \tag{3}$$

$$\epsilon_{k,i,r} = \frac{n_{ir}^k + \gamma_r}{\sum_{r'=1}^{V} n_{ir'}^k + \gamma_{r'}} \tag{4}$$

Alg. 1 shows the pseudocode for the inference phase: the Gibbs Sampling procedure starts with a random initialization; then topic assignments are estimated till convergence or till the number of performed iteration reaches the maximum value. Hyperparameters can be updated employing iterative approximation (see [15] for further details). The convergence criteria checks whether the increase in likelihood (measured on held-out data) is above a predefined threshold.

4. EXPERIMENTAL EVALUATIONS

In this section we comparatively evaluate the recommendation performance of Bayesian UCM. The experiments are aimed at assessing the quality of the proposed model in two different perspective:

- From the *forced-prediction* viewpoint, we show that the *predictive accuracy* (i.e., the prediction error) exposed by the Bayesian UCM over unobserved ratings is comparable to other state-of-the art probabilistic approaches.
- Conversely, from the *free-prediction* viewpoint, we show that Bayesian UCM is the top-notch approach in term of *recommendation accuracy*, i.e., the accuracy of the recommendation list generated.

Algorithm 1 The Gibbs-sampling procedure for parameter estimation within Bayesian UCM

Require: The sets $\mathcal{U} = \{u_1, \ldots, u_M\}$ and $\mathcal{I} = \{i_1, \ldots, i_N\}$ the rating matrix $\mathbf{R}$,the number of latent topics K, initial hyperparameters $\vec{\alpha}$, $\vec{\beta}$ and $\vec{\gamma}$.

```
initializeTopicAssignments() {Randomly assign topics}
iteration ← 0
converged ← false
while iteration < nMaxIterations and ¬converged do
    for all ⟨u, i, r⟩ ∈ R do
        z'_{u,i} ← sampleTopic(u, i, r) {According to Eq. 1};
        update counts using the new topic for the observation ⟨u, i, r⟩
    end for
    updateHyperParams()
    if (iteration > burnin) and (iteration%sampleLag = 0) then
        sampleUserTopicsMixingProbabilities() {According to Eq. 2 };
        sampleItemSelectionProbabilities() {According to Eq. 3 };
        sampleRatingProbabilities() {According to Eq. 4 };
        converged ← checkConvergence()
    end if
    iteration ← iteration + 1
end while
```

According to the empirical results originally found in [7] and subsequently confirmed in [3], there is no monotonic relationship between prediction error (or, equivalently, accuracy) and recommendation accuracy. Therefore, a low prediction error does not necessarily imply a satisfactory recommendation performance. The latter is better evaluated in terms of the accuracy of the recommendation lists provided to the users. Therefore, the findings in this section will state the superiority of the Bayesian UCM in providing accurate recommendations.

We perform the above evaluations on two reference benchmark data sets, namely MovieLens-1M[1] and a sample of Netflix data. The main features of these datasets are summarized in the table below:

	Nextflix		MovieLens	
	Training Set	Test Set	Training Set	Test Set
Users	435,656	389,305	6,040	6,040
Items	2,961	2,961	3,706	3,308
Ratings	5,714,427	3,773,781	800,168	200,041
Avg ratings (user)	13.12	9.69	132,47	33,119
Avg ratings (item)	1929.90	1274.50	215.91	60.47
Sparseness Coeff	0,9956		0,9643	

As far as Predictive Accuracy is concerned, Bayesian UCM provides the following estimation for user preference:

$$P(R = r|u, i) = \sum_z P(z|\vec{\vartheta}_u) P(r|z, i) = \sum_k \vartheta_{u,k} \epsilon_{k,i,r}$$

We evaluate the *RMSE* of Bayesian UCM over the MovieLens data set and compare its predictive accuracy against

[1]http://www.grouplens.org/system/files/million-ml-data.tar_0.gz

a selection of state-of-art probabilistic competitors, namely, Mixture of Multinomials [14], G-PLSA [9], URP-Boosted [13], URP-Gibbs [2], UCM [4] and PMF [16]. Results are summarized in Tab. 2, wherein column #**Topics** indicates the number of latent factors taken into account within each individual probabilistic model. The minimum *RMSE* value is highlighted in bold.

Approach	Best RMSE	#Topics
Mixture of Multinomials	0.9328	4
G-PLSA	0.9241	10
URP-Boosted (Variational)	0.9235	4
URP-Gibbs	0.8997	9
PMF	**0.8655**	10
UCM (Multinomial)	0.9638	4
UCM (Gaussian)	0.9359	2
Bayesian UCM	0.9263	30

Table 2: Summary of predictive competitor accuracy over the MovieLens dataset

Empirical evidence reveals that the predictive accuracy of Bayesian UCM is lower than that of G-PLSA, URP-Boosted, URP-Gibbs and PMF. This is not a surprising finding, since the generative process of the foresaid competitors is focused on the prediction accuracy. By looking at the results in Tab. 2, Bayesian UCM is superior to both the variants of UCM. In particular, Bayesian UCM outperforms significantly the UCM variant equipped with multinomial rating distribution.

It is worth providing an insight into the difference in predictive accuracy between URP-Gibbs and Bayesian UCM, since both are Bayesian probabilistic approaches based on a Gibbs sampling procedure for approximated model inference. The observed *RMSE* discrepancy is essentially due to the nature of the underlying mathematical models. Indeed, URP-Gibbs is a forced-prediction approach meant to increase the likelihood of those communities, in which a similar rating behavior is observed across the respective users. This is clearly preferable for predictive accuracy. Instead, Bayesian UCM is a free-prediction approach, that considers not only the rating behavior but also the frequency of item selection in the identification of user communities. As a matter of fact, the generic community gathers those users who tend to assign similar ratings to items that are frequently experienced within the same community. Therefore, as it has been already anticipated, an item which has received high ratings from few users of a community could not have better chances of being recommended with respect to a popular item within the same community, which has received only ratings around the average. In other words, combining rating behavior (which is the only component in forced-prediction) with item selection for free prediction tends to have a negative impact on the resulting predictive accuracy.

As already discussed, the results are significanlty different when recommendation accuracy is taken into account. Here, Bayesian UCM is compared against a selection of heterogeneous competitors, namely Top-Pop and Item-Avg [7], Pure-SVD [7], LDA [6], PLSA [11], URP-Gibbs and UCM. Item ranking in the context of the foresaid probabilistic approaches, apart from UCM, exclusively relies on item selection. The UCM is the only competitor that can combine both item selection and rating emission for item ranking, as it directly supports a free-prediction approach. Also, It is worth noticing that, though being the top-performer in terms of predictive accuracy, PMF is not considered here, because previous tests performed in [3] have shown that its recommendation accuracy is low.

The results are summarized in Fig. 2 in which Bayesian UCM is referred to as BUCM for convenience. The latter achieves the best performance against all the competitors, and in general the two datasets confirm the same trend. [2] Notice that, the performances of both Pure-SVD and PLSA over the Netflix data set are very close to Top-Pop.

The graphs show the results achieved by the selected competitors over the MovieLens data set, when the size of their recommendation lists varies from 1 to 20. It is evident that all probabilistic approaches, with the only exception of PLSA, outperform both the baseline methods, namely Top-Pop and Item-Avg, as well as Pure-SVD. This confirms the effectiveness of probabilistic modeling, as it is claimed in [3]. In particular, Bayesian UCM outperforms all competitors both in (standard and US) precision and recall.

The gain in accuracy with respect to LDA is more significant when US-precision and US-recall are taken into account (recall 0.5 vs 0.468 when $k = 20$), mainly because in this test item ranking benefits from the component of predicted relevance.

Notably, the discrepancy between the recommendation accuracy of Bayesian UCM and UCM is consistently large. This confirms the advantages of the Bayesian approach. Also, it is worth noticing how URP, though exhibiting a higher predictive accuracy than Bayesian UCM, poorly performs in terms of recommendation accuracy with respect to the latter. Such an empirical evidence confirms the importance of the selection component in the recommendation process.

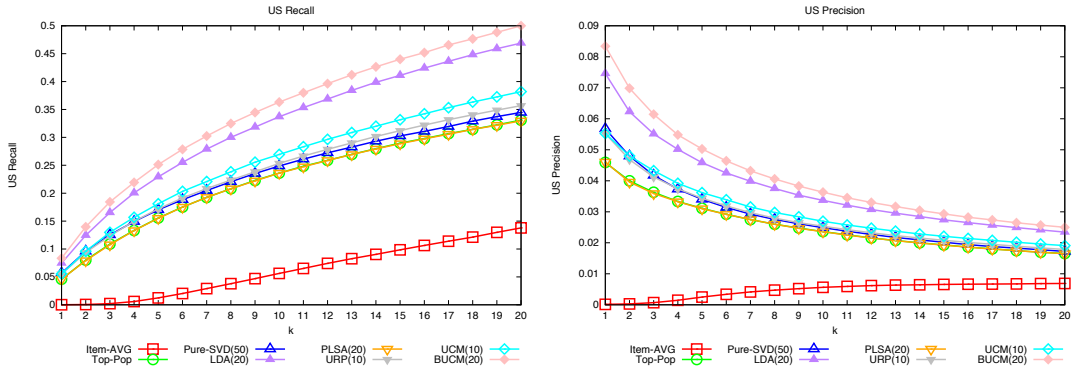
The US precision and recall of Bayesian UCM are further (comparatively) investigated over the Movielens data set, when the size of the random sample of candidate recommendations is varied in the testing protocol (recalled in section 2) from 250 to 1000. The results shown in Fig. 3 prove the superiority of Bayesian UCM.

Finally, it is important to evaluate whether the Bayesian UCM introduced a significant performance degradation. In principle, the increase in recommendation accuracy comes at a cost of a more complex model which in turn provides a more complex inference procedure. In fig. 4 we compare the execution times of the Bayesian UCM to those of the URP model. The two models exhibit a similar generative process, the difference lying in the explicit modeling (and inference) of the item selection component. The latter difference however, only yields a reasonable overhead.

5. CONCLUSIONS AND FUTURE WORK

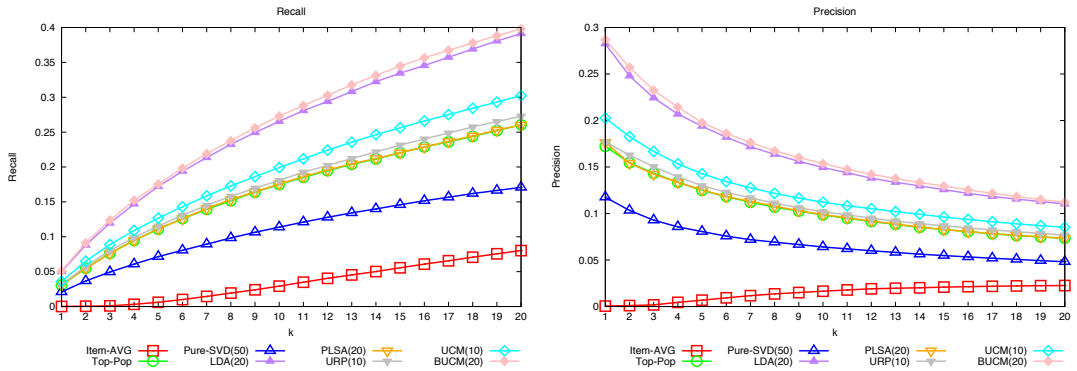
We proposed Bayesian UCM, a new probabilistic approach to recommendation, that is derived as a Bayesian reformulation of the former UCM approach [5]. Bayesian UCM is better suited to the sparsity of the rating matrix and less susceptible to overfitting. Its underlying idea is to assume a generative process, that takes into account both item selection and rating emission for the purpose of gathering those users who experience the same items and tend to adopt the same rating pattern into communities. The individual users are modeled as a random mixture of topics, where each topic is characterized by a distribution modeling the pop-

[2]We omitted some models on NetFlix data to ease readability of the overlapping curves.



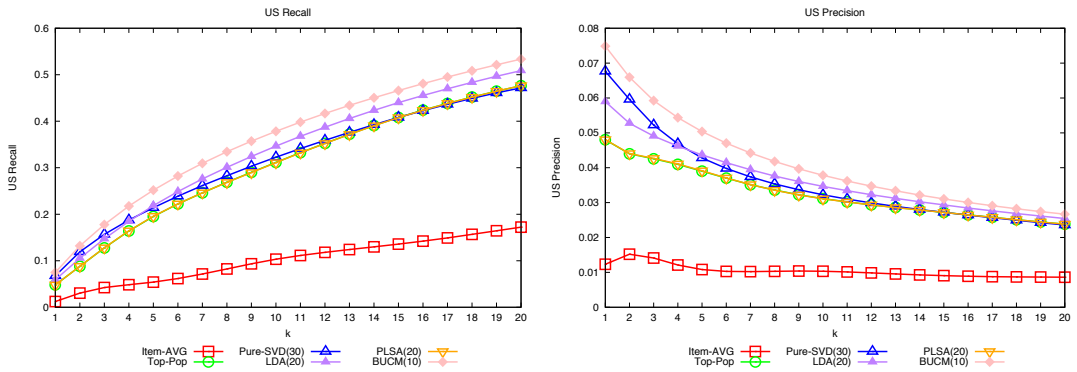
(a) US-Recall on MovieLens

(b) US-Precision on MovieLens



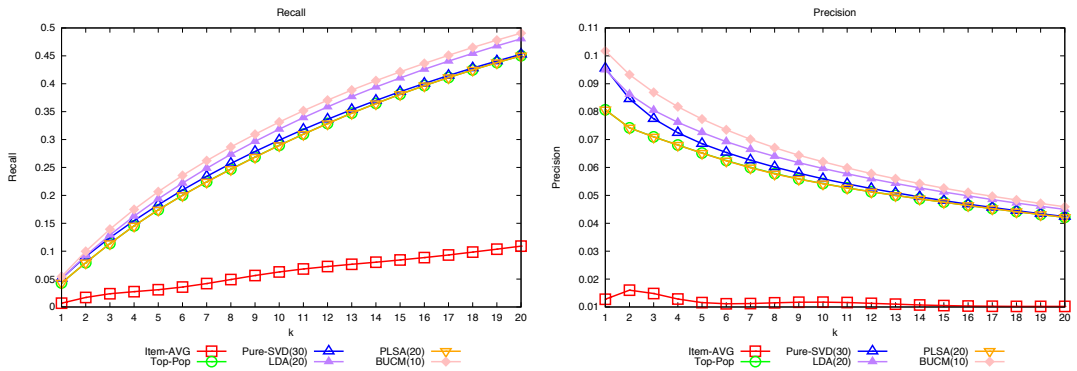
(c) Recall on MovieLens

(d) Precision on MovieLens



(e) US-Recall on NetFlix

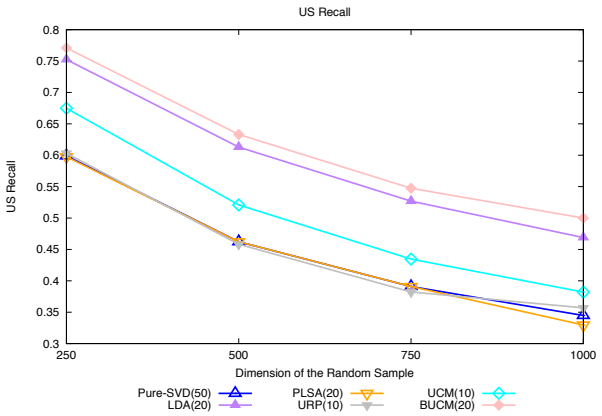
(f) US-Precision on NetFlix



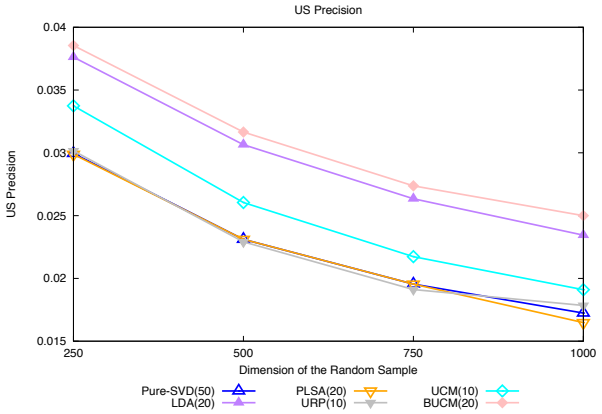
(g) Recall on NetFlix

(h) Precision on NetFlix

Figure 2: Precision and recall over the MovieLens and Netflix data sets



(a) US-Recall



(b) US-Precision

Figure 3: Recommendation accuracy over the MovieLens data set with the increasing size of the random sample and $K = 20$

ularity of items within the respective user-community and by a distribution over preference values for those items. A novel criterion for item ranking was also proposed within Bayesian UCM that accommodates both the components for item popularity within the individual user communities and rating prediction. Model inference relies on a simple and elegant procedure for approximated parameter estimation based on Gibbs sampling. An intensive experimental validation showed that Bayesian UCM outperforms state-of-art approaches to recommendation in terms of recommendation accuracy.

We planned to comparatively investigate the behavior of Bayesian UCM on larger-scale data sets such as Yahoo!!Music[3]. Moreover, ongoing research efforts aim to incorporate both collaborative and content features [1, 12, 17] within Bayesian UCM. This is expected to produce more accurate recommendations even in the case of new users/items.

6. REFERENCES

[1] D. Agarwal and B.-C. Chen. flda: matrix factorization through latent dirichlet allocation. In *Proc. WSDM Conf.*, pages 91–100, 2010.

[3]http://kddcup.yahoo.com/datasets.php

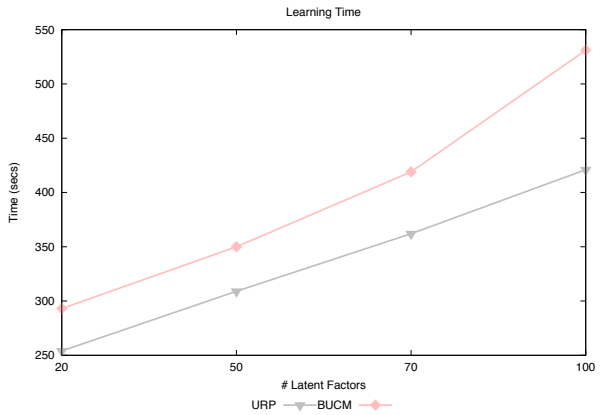


Figure 4: Execution time: BUCM vs. URP on MovieLens.

[2] N. Barbieri. Regularized gibbs sampling for user profiling with soft constraints. In *Proc. ASONAM Conf.*, 2011.

[3] N. Barbieri and G. Manco. An analysis of probabilistic methods for top-n recommendation in collaborative filtering. In *Proc. ECML-PKDD Conf.*, 2011.

[4] N. Barbieri, E. Ritacco, and G. Manco. A probabilistic hierarchical approach for pattern discovery in collaborative filtering data. In *Proc. SDM Conf.*, 2011.

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

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[7] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proc. ACM RecSys Conf.*, pages 39–46, 2010.

[8] G. Heinrich. Parameter Estimation for Text Analysis. Technical report, University of Leipzig, 2008.

[9] T. Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proc. ACM SIGIR Conf.*, 2003.

[10] T. Hofmann. Latent semantic models for collaborative filtering. *ACM TOIS*, 22(1):89–115, 2004.

[11] T. Hofmann and J. Puzicha. Latent class models for collaborative filtering. In *Proc. IJCAI Conf.*, pages 688–693, 1999.

[12] X. Jin, Y. Zhou, and B. Mobasher. A maximum entropy web recommendation system: combining collaborative and content features. In *Proc. KDD Conf.*, pages 612–617, 2005.

[13] B. Marlin. Modeling user rating profiles for collaborative filtering. In *Proc. NIPS Conf.*, 2003.

[14] B. Marlin. Collaborative filtering: A machine learning perspective. Technical report, Department of Computer Science University of Toronto, 2004.

[15] T. P. Minka. Estimating a dirichlet distribution. Technical report, MIT, 2003.

[16] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Proc. NIPS Conf.*, pages 1257–1264, 2008.

[17] D. H. Stern, R. Herbrich, and T. Graepel. Matchbox:

large scale online bayesian recommendations. In *Proc. WWW Conf.*, pages 111–120, 2009.