Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Balancing Prediction and Recommendation Accuracy: Hierarchical Latent Factors for Preference Data

Nicola Barbieri[1], Giuseppe Manco[2],
Ettore Ritacco[2], Riccardo Ortale[2]

RT-ICAR-CS-11-06 **Ottobre 2011**

1 Università Della Calabria – Dipartimento di Elettronica Informatica e Sistemistica (DEIS) – Via P. Bucci 41C, 87036 Rende, Italy, URL: *www.unical.it*

2 Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR) – Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: *www.icar.cnr.it*

Balancing Prediction and Recommendation Accuracy: Hierarchical Latent Factors for Preference Data

Nicola Barbieri * Giuseppe Manco † Ettore Ritacco‡ Riccardo Ortale§

Abstract

Recent works in Recommender Systems (RS) have investigated the relationships between the prediction accuracy, i.e. the ability of a RS to minimize a cost function (for instance the RMSE measure) in estimating users' preferences, and the accuracy of the recommendation list provided to users. State-of-the-art recommendation algorithms, which focus on the minimization of RMSE, have shown to achieve weak results from the recommendation accuracy perspective, and vice versa. In this work we present a novel Bayesian probabilistic hierarchical approach for users' preference data, which is designed to overcome the limitation of current methodologies and thus to meet both prediction and recommendation accuracy. According to the generative semantics of this technique, each user is modeled as a random mixture over latent factors, which identify users community interests. Each individual user community is then modeled as a mixture of topics, which capture the preferences of the members on a set of items. We provide two different formalization of the basic hierarchical model: *BH-Forced* focuses on rating prediction, while *BH-Free* models both the popularity of items and the distribution over item ratings. The combined modeling of item popularity and rating provides a powerful framework for the generation of highly accurate recommendations. An extensive evaluation over two popular benchmark datasets, proves the effectiveness and the quality of the proposed algorithms, showing that *BH-Free* realizes the best compromise between prediction and recommendation accuracy.

Keywords. Recommender Systems, Probabilistic Hierarchical Co-clustering, Recommendation Accuracy.

1 Introduction

Recommender systems (RS) play an important role in several domains as they provide users with potentially interesting recommendations within catalogs of available information/products/services [19]. Among the various RS techniques, Collaborative Filtering (CF) is effective with huge catalogs when information about past interactions is available. According to this assumption, several CF-based recommendation techniques have been proposed, mainly focusing on the predictive skills of the system.

Recent studies [8, 9, 17] have shown that the focus on prediction does not necessarily help in devising good recommender systems. In particular, the improvements in prediction accuracy do not automatically reflect into improvements of the accuracy of the recommendation list, which is actually displayed to users. It has been shown [3, 4] that probabilistic approaches based on latent-factor models allow the most adequate degree of flexibility, as they: *(i)* allow the specification of complex yet easy to interpret latent structures; *(ii)* achieve the highest recommendation accuracy.

Typically, complex patterns can be better detected by means of co-clustering approaches [3, 10, 18, 22–24]. The latter aim at partitioning data into homogeneous blocks enforcing a simultaneous clustering on both the dimensions of the preference data. This highlights the mutual relationships between users and items. The work in [5] further extends the co-clustering approaches by proposing the Hierarchical User Community Model (HUCM in the following), which overcomes the limits of a static structure enforced by fixed row/column blocks where both users and items have to fit. HUCM introduces a dynamic hierarchy between user communities and item categories: in practice, data are modeled assuming that there is a dependency relationship between latent factors on items and latent factors on users.

When focusing on user communities only, HUCM is incidentally capable of explicitly modeling item selection, i.e, the probability that an item is actually selected by a user. 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 non-hierarchical version of HUCM (referred to as UCM in the following) is capable to model item selection and rating prediction simulta-

*ICAR-CNR and DEIS, Univ. of Calabria, Italy. E-mail: barbieri@icar.cnr.it

†ICAR-CNR, Italy. E-mail: manco@icar.cnr.it

‡ICAR-CNR, Italy. E-mail: ritacco@icar.cnr.it

§ICAR-CNR, Italy. E-mail: ortale@icar.cnr.it

neously.

To summarize, previous research devised two major contributions to the current literature. First, hierarchical probabilistic structures based on latent factor models can better model the underlying hidden relationships at the basis of users' behaviors. This allows to boost the prediction accuracy of such probabilistic models. Second, explicit modeling of item selection plays a crucial role with accurate recommendation lists. As shown in [4], a combined use of items selection and ranking prediction is crucial for providing accurate recommendation lists.

There is an apparent mismatch between these two situations. The explicit modeling of item selection boosts the accuracy of recommendation lists, yet it negatively impacts on prediction accuracy. The point is that exploiting item selection for ranking prediction in a (hierarchical) co-clustering model yields too many parameters to estimate, and consequently the risk of overfitting increases. As a matter of fact, the models achieving better prediction accuracy [1, 5, 18, 20] ignore the item selection components, whereas the models exhibiting the highest recommendation accuracy (such as *Pure-SVD* [9], *pLSA* [14], *LDA* [7] and *UCM*) provide poor performance in ranking prediction, or do not support it at all.

In this paper we propose a new Bayesian Hierarchical latent factor model (*BH* in the following) which combines the advantages of both hierarchical modeling and item selection, and comparatively investigate both its recommendation accuracy and prediction error. BH relies on a generative process, which can take 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. Individual users are modeled as a random mixture of communities, where the individual community is characterized again by a mixture of topics modeling both the popularity of items and the distribution over item ratings.

BH reinterprets the former HUCM in a Bayesian modeling setting, that is better suited to the sparseness of the preference data and less susceptible to overfitting. Additionally, *BH* allows a simpler and more elegant procedure for the estimation of model parameters through Gibbs sampling [6]. As a matter of fact, a reinterpretation of some results in [5] has been initially studied in [2]. There, we proposed the *Bayesian User Community Model* (BUCM), which revises the UCM model in a Bayesian settings. Again, BUCM exhibits the (so far) highest recommendation accuracy, but still fails in providing a suitable trade-off with prediction accuracy. By converse, the *BH* model proposed here represents a systematic accommodation of the above issues, as it meets the aforementioned requirements in a simple and elegant mathematical setting, which guarantees both recommendation and prediction accuracy.

The rest of the paper is organized as follows. First, we give an overview of the recommendation problem by introducing some preliminary notations in Sec. 2. In Sec. 3 we introduce and discuss two versions of the hierarchical model, which focus respectively on ranking prediction and explicit modeling of item popularity. A collapsed Gibbs sampling procedure for parameter estimation is also specified. We evaluate the proposed approaches in Sec. 4, showing that the BH approach outperforms state-of-the-art competitors in recommendation accuracy and is yet comparable to them in terms of prediction error. Finally, conclusions are drawn in Sec. 5.

2 Preliminaries and Context

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. Users' 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 value r_i^u falls 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 $\overline{r}_{\mathbf{R}}^u$ denotes the average rating among all those ratings $r_i^u > 0$ of the user u.

The number of users M as well as the number of items N are very large 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. Finally, the number of pairs $(u, i) \in \mathbf{R}$ such that $r_i^u > 0$ is denoted as $\mathcal{S}$.

Given an active user u, the goal of a RS is to provide u with a recommendation list $\mathcal{RL}_u \subseteq \mathcal{I}$ of unexperienced items (i.e., $\mathcal{RL}_u \cap \mathcal{I}_{\mathbf{R}}(u) = \emptyset$), that are expected to be of interest to u. This clearly involves predicting the interest of u into unrated items.

In this paper, we focus on probabilistic approaches based on latent factors. In these models, each preference observation $\langle u, i\rangle$ is generated by one of multiple possible states, which informally explains the reason why u rated i. To keep notation uncluttered, we shall write $P(r, u, i)$ to denote the joint probability $P(R = u, U = u, I = i)$, 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, for instance $P(r|u,i)$ 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 [14]:

- *Forced prediction*: the probabilistic model provides an estimate of $P(r|u,i)$;
- *Free prediction*: the item selection process is included in the probabilistic model, which is typically based on the estimate of $P(r,i|u)$. Since the latter can be factorized as $P(r|i,u)P(i|u)$, the resulting model still includes a forced prediction component, which however is weighted by the item selection component.

In general, a recommendation list $\mathcal{RL}_u$ 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 in i.
- Sort $\mathcal{C}$ in descending order of item scores p_i^u;
- Add the first k items from $\mathcal{C}$ to $\mathcal{RL}_u$ and return the latter to user u.

Historically, the evaluation of the goodness of the recommendation list is made implicitly, i.e. by reinterpreting the recommendation problem as a missing value prediction problem [21]. Since a user is more prone to access items for which she will likely provide a positive feedback, a recommendation list can be be built by drawing upon the (predicted) highly-rated items. Under this perspective, predictive accuracy metrics measure how close the predicted score are to the true user preferences [12], typically through the *Root Mean Square Error* (RMSE).

However, the interaction between the RS and the user is often based exclusively on the recommendation list, while the system does not directly provide predicted rating to users. In this context, classification accuracy metrics, such as precision and recall, are more suitable to measure the effectiveness of the RS.

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 recommendation list $\mathcal{RL}_u$ is the set of the best items drawn from $\mathcal{I} - \mathcal{I}_{\mathbf{T}}(u)$. Evaluation is performed by comparing $\mathcal{RL}_u$ with $\mathcal{I}_{\mathbf{S}}(u)$. Given a user u and a subset $\mathcal{T}_u^r \subseteq \mathcal{I}_{\mathbf{S}}(u)$ of relevant items, the degree of precision (*prec*) and recall (*rec*) of the k items within $\mathcal{RL}_u$ is defined as shown next:

$$rec(k) = \frac{1}{M}\sum_{u=1}^{M}\frac{|\mathcal{RL}_u \cap \mathcal{T}_u^r|}{|\mathcal{T}_u^r|}$$

$$prec(k) = \frac{1}{M}\sum_{u=1}^{M}\frac{|\mathcal{RL}_u \cap \mathcal{T}_u^r|}{k}$$

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:

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

The above definitions of precision and recall consider the amount of useful recommendations as a single session. A different perspective can be considered by assuming that a recommendation meets *user satisfaction*, if the user can find at least a *hit*, i.e. an interesting (best rated) item in the recommendation list. Starting from a redefinition of the set of relevant items,

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

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

- For each user u and for each item $i \in \mathcal{T}_u^{\prime r}$:
  - 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*.

According to this protocol, [9] defines the *US-Precision* and *US-Recall*.

$$\textit{US-Recall}(k) = \frac{\#hits}{|\mathcal{T}_u^{\prime r}|}, \quad \textit{US-Precision}(k) = \frac{\#hits}{k \cdot |\mathcal{T}_u^{\prime r}|}$$

A key role in the process of generating accurate recommendation lists is played by the schemes with which to rank items candidate for recommendation. [4] 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 be summarized as follows.

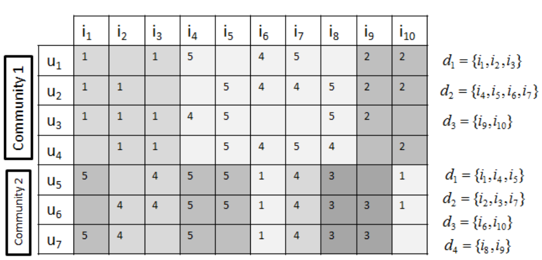
- Lower RMSE values do not necessarily imply improvements in recommendation accuracy. Cutting-edge probabilistic approaches, such as PMF [20], equipped with *expected-value* ($p_i^u = E[R|u, i]$) item-ranking schemes have been shown to perform poorly in terms of recommendation accuracy.

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

In the model proposed in this paper, we shall concentrate on a mix of *item selection and relevance ranking*, namely $p_i^u = P(i, r > \overline{r}_{\mathbf{T}}^u|u)$. Specifically, 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.

3 Bayesian Hierarchical Model for Preference Data

A crucial point in the foregoing discussion is the observation that different communities can infer different evaluations of the same item. The problem has been preliminarily studied in [5], where the concepts of user communities and hierarchical item categories was introduced. Specific groups of users tend to be co-related according to different subsets of features.



		i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}	
Community 1	u_1	1		1	5		4	5		2	2	$d_1 = \{i_1, i_2, i_3\}$
	u_2	1	1			5	4	4	5	2	2	$d_2 = \{i_4, i_5, i_6, i_7\}$
	u_3	1	1	1	4	5			5	2		$d_3 = \{i_9, i_{10}\}$
	u_4		1	1		5	4	5	4		2	
Community 2	u_5	5		4	5	5	1	4	3		1	$d_1 = \{i_1, i_4, i_5\}$
	u_6		4	4	5	5	1	4	3	3	1	$d_2 = \{i_2, i_3, i_7\}$
	u_7	5	4		5		1	4	3	3		$d_3 = \{i_6, i_{10}\}$ $d_4 = \{i_8, i_9\}$

Figure 1: Hierarchical topics nested into user communities.

Consider the toy example in fig. 1, where homogeneous blocks exhibiting similar rating patterns are highlighted. There are 7 users clustered into two main communities. Community 1 is characterized by 3 main topics (with groups $d_{11} = \{i_1, i_2, i_3\}$, $d_{12} = \{i_4, i_5, i_6, i_7\}$ and $d_{13} = \{i_8, i_9, i_{10}\}$), whereas community 2 includes 4 main topics (with groups $d_{21} = \{i_1, i_4, i_5\}$, $d_{22} = \{i_2, i_3, i_7\}$, $d_{23} = \{i_6, i_{10}\}$ and $d_{24} = \{i_8, i_9\}$). The novelty is that different communities group the same items differently. This introduces a topic hierarchy which in principle increases the semantic power of the overall model.

In this paper we extend the framework of [5] by relaxing some basic conditions:

- users can exhibit diverse "dynamic" behaviors (in the style of [15]). That is, for each user there is no fixed community. Rather, the local behavior is picked randomly among the most probable.

- Analogously, items are dynamically associated with topics according to an underlying probability law.

- The overall process is governed by Bayesian priors thus allowing a more controlled modeling of data sparseness.

The key idea is that there exists a set of user communities, each one describing different tastes of users and their corresponding rating patterns. Each user community is then modeled as a random mixture over latent topics, which can be interpreted as item-categories. Given a user u, we can foresee his/her preferences on a set of items $\mathcal{I}_u$ by choosing an appropriate user community z and then choosing an item category w for each item in the list. The choice of the item category w actually depends on the selected user community z. Finally the preference value is generated by considering the preference of users belonging to the group z on items of the category w. This local modeling of items is the main difference in the generative semantic with respect to state-of-art LDA based co-clustering approaches [18].

A first coarse-grained generative process directly derived from [5] can be devised as an adaptation of the well-know LDA-based models [1, 7], and is graphically depicted in Fig. 2:

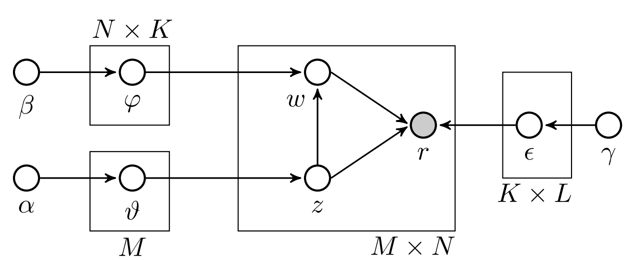


Figure 2: *BH-Forced* Generative model

1. For each user $u \in \mathcal{U}$ sample user community-mixture components $\vec{\vartheta}_u \sim Dir(\tilde{\alpha})$;

2. For each item $i \in \mathcal{I}$ and user community $z \in \{1, \ldots, K\}$ sample the mixture components $\vec{\varphi}_{z,i} \sim Dir(\vec{\beta})$

3. For each topic $w \in \{1, \ldots, L\}$ and user community $z = \{1, \cdots, K\}$, sample rating probabilities $\vec{\varepsilon}_{z,w} \sim Dir(\vec{\gamma})$

4. For each active pair $n = (u, i)$ in $\mathbf{R}$:

   (a) Choose a user attitude $z_n \sim Discrete(\vec{\vartheta}_u)$

   (b) Choose a topic $w_n \sim Multi(\vec{\varphi}_{z_n,i})$

   (c) Generate a rating value for the chosen item according to the distribution $P(r|\vec{\varepsilon}_{z_n,w_n})$

With respect to HUCM proposed in [5], that relies on maximum likelihood estimation with multinomial priors for model inference, the new Bayesian formulation (*BH-Forced* in the following) 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 [6]. Notice that, in the following, we model $P(r|\vec{\varepsilon}_{z_n,w_n})$ as a multinomial over the parameter vector $\vec{\varepsilon}_{z_n,w_n}$. Different choices can be made, in the style of [13], which are omitted here for lack of space.

Figure 3 shows how the rating matrix described in Fig. 1 can be modeled according to *BH-Forced*. The figure summarizes a setting of the probability distributions for a *BH-Forced* Co-Clustering model compatible with the data represented in the previous example. By applying the generative process described above, the interested reader can easily verify that each observed rating can be replicated by drawing upon the corresponding distribution. For example, let us consider the observation $\langle u_5, i_5 \rangle$. According to the devised generative process, we first pick user community 2 for u_5, exploiting table *c*. Next, we assign item category 1 to item i_5, by drawing upon the available categories according to the probability in table *e*. Finally, given the cocluster $\langle 2, 1 \rangle$, we observe rating 5 by picking randomly according to the related rating distribution in table *f*.

Again, it is worth noticing that the Bayesian Hierarchical model is more powerful, as it allows the modeling of complex relationship in a more dynamic scenario. As a matter of fact, users (resp. items) are not necessarily statically bound to a single community (resp. topic), but their membership can be dynamically modeled. In particular, for each pair $\langle u, i \rangle$ diverse user communities and item categories can be picked, according to the associated multinomial priors.

3.1 Modeling Free Prediction. A problem with the BH model introduced so far is its focus on forced-prediction. That is, the model concentrates on the prediction of preference values for each observed user-item pair, and does not explicitly take into account item selection. As already mentioned, this component plays a crucial role in the generation of the recommendation list. Hence, it is likely to expect poor recommendation accuracy for this model.

The point is that the components in the *BH-Forced* model do not provide a direct support to the computation of $p(r, i|u)$. Thus, the only possibility for *BH-Forced* is to generate a recommendation list by resorting to the expected-value, as explained in section 2.

We fix this issue by accommodating the hierarchical schema in Fig. 2 with an explicit item selection component. Specifically, 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. In particular, the distribution of items given the topic variable w depends on the choice of the user community: this enforces an explicit modeling of item popularity both within a category and within a community, and hence provides a high degree of flexibility. Further, the rating prediction components maintains almost the same structure as in the *BH-Forced* models, and hence even the accuracy is almost the same.

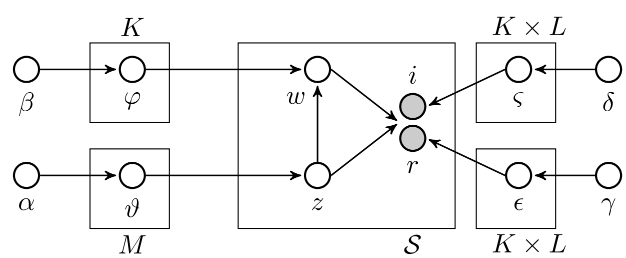


Figure 4: *BH-Free* Model

The generative process for the new *BH-Free* model, whose corresponding graphical scheme is shown in Fig. 4, is as follows:

1. For each user $u \in \mathcal{U}$ sample user community-mixture components $\vec{\vartheta}_u \sim Dir(\tilde{\alpha})$;

2. For each user community $z \in \{1, \ldots, K\}$ sample the mixture components $\vec{\varphi}_z \sim Dir(\vec{\beta})$

3. For each topic $w \in \{1, \ldots, L\}$ and user community $z = \{1, \cdots, K\}$,

Items

Users	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	1		1	1		1	1		1	1
u_2	1	1			1	1	1	1	1	1
u_3	1	1	1	1	1			1	1	
u_4		1	1		1	1	1	1		1
u_5	2		2	2	2	2	2	2		2
u_6		2	2	2	2	2	2	2	2	2
u_7	2	2		2		2	2	2	2	

(a) User-Topic Assignments

Items

Users	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	1		1	2		2	2		3	3
u_2	1	1			2	2	2	2	3	3
u_3	1	1	1	2	2			2	3	
u_4		1	1		2	2	2	2		
u_5	1		2	1	1	3	2	4		3
u_6		2	2	1	1	3	2	4	4	3
u_7	1	2		1		3	2	4	4	

(b) Item-Category Assignments

Topic

Users	1	2
u_1	0.9	0.1
u_2	0.95	0.05
u_3	0.9	0.1
u_4	1	0
u_5	0.05	0.95
u_6	0.02	0.98
u_7	0.1	0.9

(c) User-Topic Prob

Items

Categories	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
1	0.98	0.95	0.9	0	0.03	0	0.03	0	0	0.02
2	0.02	0.05	0.1	0.9	0.91	0.99	0.97	0.95	0.1	0
3	0	0	0	0.1	0.06	0.01	0	0.05	0.9	0.98
4	0	0	0	0	0	0	0	0	0	0

(d) Item-Category Prob - Topic 1

Items

Categories	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
1	0.9	0.05	0.01	0.95	0.98	0.03	0	0.05	0.01	0
2	0.05	0.93	0.99	0.01	0.01	0	0.95	0.15	0	0
3	0	0.01	0	0.04	0	0.97	0.04	0	0	0.99
4	0.05	0.01	0	0	0.01	0	0.01	0.80	0.99	0.01

(e) Item-Category Prob - Topic 2

		User Topic	
Rating	Item Cat	1	2
1	1	0.9	0
	2	0	0
	3	0	0.98
	4	0	0
2	1	0.01	0
	2	0	0
	3	0.98	0.02
	4	0	0.01
3	1	0.01	0.01
	2	0	0
	3	0.02	0
	4	0	0.99
4	1	0.02	0.01
	2	0.5	0.99
	3	0	0
	4	0	0
5	1	0.06	0.98
	2	0.5	0.01
	3	0	0
	4	0	0

(f) Rating Distributions

Figure 3: Probabilistic modeling of local patterns

(a) Sample item selection components $\vec{\varsigma}_{z.w} \sim Dir(\vec{\delta})$

(b) Sample rating probabilities $\vec{\varepsilon}_{z,w} \sim Dir(\vec{\gamma})$

4. For each $u \in \mathcal{U}$

   (a) Sample the number of items for the user u, $N_u \propto Poisson(\mathcal{K})$

   (b) For $n = 1$ to N_u

      i. Choose a user attitude $z_{u,n} \sim Discrete(\vec{\vartheta}_u)$

      ii. Choose a topic $w_{u,n} \sim Multi(\vec{\varphi}_{z_n})$

      iii. Choose an item $i_n \sim Multi(\vec{\varsigma}_{z_{u,n},w_{u,n}})$

      iv. Generate a rating value for the chosen item according to the distribution $P(r|\vec{\varepsilon}_{z_{u,n},w_{u,n}})$.

BH-Free tries to infer the tendency of a user to experience some items over others independent of her/his rating values. The model assumes that this tendency is influenced by implicit 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 a high probability, although this has no impact on the rating assigned to the aforesaid item category. The probability of observing an item is independent from the rating assigned, given the state of the latent variables. This is a major difference with respect to most of the (co-clustering) models, which instead approach the problem from a *matrix approximation* perspective (as they focus on the prediction of r_i^u). By contrast, free-prediction models are focused on both the estimation of a rating behavior 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.

It is worth noticing that support to free prediction was already included in the UCM model. And in fact, *BH-Free* can be considered a substantial extension of the UCM model, in that it *(i)* adds a hierarchical co-clustering structure, thus complying to the originary idea of modeling local patterns; *(ii)* accommodates a Bayesian modeling which allows better control on data sparseness.

3.2 Inference and parameter estimation. The inference process is similar for both *BH-Forced* and *BH-Free*. Concerning the *BH-Free* model, there's a small overhead due to the explicit modeling of item selection. Hence, in the following we shall only sketch the derivation of the sampling equations for this model. The equations for *BH-Forced* can be derived by resorting to similar techniques.

SYMBOL	DESCRIPTION
M	#Users
N	# Items
$\mathbf{R}$	$M \times N$ Rating Matrix
K	# topics/user communities
L	# item categories
Θ	matrix $M \times K$ of parameters $\vec{\vartheta}_u$
$\vec{\vartheta}_u$	K-vector: mixing proportion of user-communities for the user u
Φ	Matrix of parameters $\vec{\varphi}_k$
$\vec{\varphi}_k$	L-vector: mixing proportion for the item category l and the user-topic k
Γ	Matrix of parameters $\vec{\epsilon}_{k,l}$
$\vec{\epsilon}_{k,l}$	V-vector: distribution over rating values for the co-cluster k, l
Σ	Matrix of parameters $\vec{\varsigma}_{k,l}$
$\vec{\varsigma}_{k,l}$	N-vector: mixing proportion for each item i in the co-cluster k, l
z	user-topic variable
Z	$M \times N$ matrix: user-topic assignments for each rating observation
w	item-categories topic variable
W	$M \times N$ matrix: item-categories assignments for each rating observation
$\vec{\alpha}$	K- vector: Dirichlet priors on user communities
$\vec{\beta}$	L-vector: Dirichlet priors on item categories
$\vec{\gamma}$	V-vector: Dirichlet priors on rating values
$\vec{\delta}$	N-vector: Dirichlet priors on items
n_u^k	# evaluation of the user u which have been assigned to the user topic k
$n_r^{k,l}$	# times that the rating r has been assigned to each observation when the user topic is k and the item category is l
$n_i^{k,l}$	# times that the item category l has been assigned to observations of the item i when the user topic is k
n_u	# observations for the user u ($\lvert\mathcal{I}(u)\rvert$)
n_k	# observations associated with community k
$n_{k,l}$	# times that the category l has been assigned to observations whose user topic is k
$\vec{n}_u$	$\{n_u^k\}_{k=1}^K$
$\vec{n}_k$	$\{n_k^l\}_{l=1}^L$
$\vec{n}_{k,l}^{(V)}$	$\{n_r^{k,l}\}_{r=1}^V$
$\vec{n}_{k,l}^{(N)}$	$\{n_i^{k,l}\}_{i=1}^N$

Table 1: Summary of notation

The notation used in our discussion is summarized in Tab. 1. Given the hyperparameters $\vec{\alpha}$, $\vec{\beta}$, $\vec{\delta}$ and $\vec{\gamma}$, the joint distribution of the data $\mathbf{R}$, the user-community mixtures Θ, the item-topic components Φ, the item and rating probabilities Σ and Γ and the observation-community/topic assignments Z, W, can be computed as:

$$\begin{aligned} P(\mathbf{R}, Z, W, \Theta, \Phi, \Sigma, \Gamma|\vec{\alpha}, \vec{\beta}, \vec{\gamma}, \vec{\delta}) = & \\ & P(\mathbf{R}|Z, W, \Gamma, \Sigma) \\ & \cdot P(Z|\Theta)P(\Theta|\vec{\alpha}) \\ & \cdot P(W|Z, \Phi)P(\Phi|\vec{\beta}) \\ & \cdot P(\Gamma|\vec{\gamma}) \\ & \cdot P(\Sigma|\vec{\delta}) \end{aligned} \tag{3.1}$$

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

$$\begin{aligned} P(\mathbf{R}, Z, W|\vec{\alpha}, \vec{\beta}, \vec{\gamma}, \vec{\delta}) = & \int P(Z|\Theta)P(\Theta|\vec{\alpha})\mathrm{d}\Theta \\ & \int P(W|Z\Phi)P(\Phi|\vec{\beta})\mathrm{d}\Phi \\ & \int\int P(\mathbf{R}|Z, W, \Sigma, \Gamma)P(\Sigma|\vec{\delta})P(\Gamma|\vec{\gamma})\mathrm{d}\Sigma\mathrm{d}\Gamma \end{aligned}$$

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

$$\begin{aligned} P(\mathbf{R}, Z, W|\vec{\alpha}, \vec{\beta}, \vec{\gamma}, \vec{\sigma}) = & \prod_{u=1}^{M} \frac{\Delta(\vec{n}_u + \vec{\alpha})}{\Delta(\vec{\alpha})} \cdot \prod_{k=1}^{K} \frac{\Delta(\vec{n}_k + \vec{\beta})}{\Delta(\vec{\beta})} \\ & \cdot \prod_{k=1}^{K}\prod_{l=1}^{L} \frac{\Delta(\vec{n}_{k,l}^{(V)} + \vec{\gamma})}{\Delta(\vec{\gamma})} \cdot \prod_{k=1}^{K}\prod_{l=1}^{L} \frac{\Delta(\vec{n}_{k,l}^{(N)} + \vec{\delta})}{\Delta(\vec{\delta})} \end{aligned}$$

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

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

This formula is however intractable, mainly because the computation of the denominator requires a summation over an exponential number of terms. Gibbs Sampling [6] addresses this problem by defining a Markov chain, in which at each step inference can be accomplished by exploiting the full conditional $P(z_n = k_n, w_n = l_n|Z_{\neg n}, W_{\neg n}, \mathbf{R}, \vec{\alpha}, \vec{\beta}, \vec{\gamma})$. In the latter, z_n (resp. w_n) is the cell of the matrix Z (resp. W) which corresponds to this observation, and $Z_{\neg n}$ ($W_{\neg n}$) denotes the remaining topic assignments. The chain is hence defined by iterating over the available states n. The Gibbs

Sampling algorithm estimates the probability of assigning the pair k_n, l_n to the n-th observation, given the assignment corresponding to all the other rating observations:

(3.2)

$$P(z_n = k_n, w_n = l_n | Z_{\neg n}, W_{\neg n}, \mathbf{R}, \vec{\alpha}, \vec{\beta}, \vec{\gamma}) \propto \frac{n_{u_n}^{k_n} + \alpha_{k_n} - 1}{\sum_{k'=1}^{K} (n_{u_n}^{k'} + \alpha_{k'}) - 1} \cdot \frac{n_{k_n}^{l_n} + \beta_{l_n} - 1}{\sum_{l'=1}^{L} (n_{k_n}^{l'} + \beta_{l'}) - 1} \cdot \frac{n_{r_n}^{k_n, l_n} + \gamma_{r_n} - 1}{\sum_{r=1}^{V} (n_r^{k_n, l_n} + \gamma_r) - 1} \cdot \frac{n_{i_n}^{k_n, l_n} + \gamma_{i_n} - 1}{\sum_{i=1}^{N} (n_i^{k_n, l_n} + \delta_i) - 1}$$

Given the state of the Markov chain, denoted my $\mathcal{M} = (\mathbf{R}, Z, W)$, we can obtain the multinomial parameters Φ and Θ and Γ noticing that, by applying Bayes's rule and then by algebraic manipulations and the properties of the Dirichlet distribution [11]. This ultimately yields the following estimations:

$$\vartheta_{u,k} = \frac{n_u^k + \alpha_k}{n_u + \sum_{k=1}^{K} \alpha_k} \tag{3.3}$$

$$\varphi_{k,l} = \frac{n_l^k + \beta_l}{n_k + \sum_{l=1}^{L} \beta_l} \tag{3.4}$$

$$\epsilon_{k,l,r} = \frac{n_r^{k,l} + \gamma_r}{n_{k,l} + \sum_{r'=1}^{V} \gamma_{r'}} \tag{3.5}$$

$$\varsigma_{k,l,i} = \frac{n_i^{k,l} + \delta_i}{n_{k,l} + \sum_{i'=1}^{N} \delta_{i'}} \tag{3.6}$$

Finally, given the pair $\langle u, i \rangle$ we compute the probability of observing the rating value r in a *free prediction* context:

$$p(R = r, i|u) = \sum_{k=1}^{K} \sum_{l=1}^{L} \vartheta_{u,k} \cdot \varphi_{k,l} \cdot \varsigma_{k,l,i} \cdot \epsilon_{k,l,r} \tag{3.7}$$

Notice the explicit reference, in Eq. 3.7 to the $\varsigma_{k,l,i}$ components that models the probability of i being selected within co-cluster k, l. Clearly, such a component biases the ranking towards relevant items, thus providing the required adjustment that makes the model suitable for both prediction and recommendation accuracy.

4 Evaluation

In this section we comparatively evaluate the performance of the two BH models. The experiments are aimed at assessing the quality of the models in two different perspectives:

- From the *forced-prediction* viewpoint, we show that the *predictive accuracy* (i.e., the prediction error) exposed by both the *BH-Forced* and *BH-Free* models over unobserved ratings is comparable and in some cases even better to other state-of-the art probabilistic approaches.
- Conversely, from the *free-prediction* viewpoint, we show that *BH-Free* is the top-notch approach in term of *recommendation accuracy*

We use two reference benchmark data sets, namely MovieLens-1M[1] and a sample of Netflix data. Both datasets contain explicit preference data: ratings fall within the range 1 to 5, where the latter denotes the highest preference value. The main features of these datasets are summarized in Tab. 2.

	Nextflix		MovieLens	
	Training Set	Test Set	Training Set	Test Set
Users	435,656	389,305	6,040	6,032
Items	2,961	2,961	3,706	3,444
Ratings	5,714,426	3,773,781	800,729	199,480
Avg ratings (user)	13	9	132	33
Avg ratings (item)	1929	1274	216	57
Min ratings (user)	1	1	11	1
Min ratings (item)	5	1	1	1
Max ratings (user)	957	691	1849	465
Max ratings (item)	64492	42780	2738	690
Sparseness Coeff	0,9957		0,9642	
% of *	4.55	4.53	5.62	5.58
% of **	10.06	10.06	10.76	10.74
% of ***	28.82	28.87	26.11	26.11
% of ****	33.33	33.39	34.89	34.89
% of *****	23.21	23.13	22.61	22.69

Table 2: Summary of the Data used for validation.

We compare both models with some state-of-the-art competitor approaches to CF recommendation, and in particular with co-clustering approaches. For the latter aspect, we compare with *LDCC* [25] (which extends the Bayesian co-clustering model proposed in [22] and it is based on a collapsed Gibbs sampling algorithm to perform parameter estimation and inference); with *Bregman-CC* proposed in [10] (which is based on the Bregman co-clustering algorithm); with *Bi-LDA* [18] (which extends the standard URP model [1] in both the user and item dimensions). All models have been trained by retaining the 1% of the training data as held out to perform *early stopping* and avoid overfitting.

We also compare with the User community models previously defined: UCM, HUCM [5], and BUCM [2]. Whereas HUCM is a natural choice for comparison, (as the BH models represent a direct extension of such a model), the UCM (and its Bayesian redefinition) explicitly model item selection and relevance ranking, and hence represent a reference comparison for the *BH-Free* model.

[1] `http://www.grouplens.org/system/files/million-ml-data.tar\_\_0.gz`

Predictive Accuracy. We start our analysis from the evaluation of the prediction accuracy achieved by the algorithms. Table 3 summarizes the best RMSE obtained on both the considered datasets, together with the associated settings. To assess the effectiveness of all the considered approaches in rating prediction, we compare them with *Probabilistic Matrix Factorization (PMF)* [20], a cutting-edge probabilistic approach.

As a general remark, both *BH-Forced* and *BH-Free* exhibit similar RMSE as other co-clustering approaches. *BH-Free* even outperforms all the other approaches on the NetFlix data, and is the runner-up winner after *HUCM* which, however, exhibits a marginal advantage. Minimal differences can also be noticed on MovieLens, where *PMF* achieves the best RMSE score (as expected). In both datasets, *BUCM* is overcome by all other co-clustering methods: this proves that a hierarchical structure provides substantial information for boosting the accuracy of prediction.

Since the dependency between item categories and user communities tends to produce more complex structures with respect to traditional co-clustering approaches, it is important to evaluate the scalability of the *BH* models in this respect. Fig. 5 shows how the RMSE scales with the number of item categories for the two *BH* models. *BH-Forced* globally achieves a lower RMSE, but tends to overfit the data with a larger number of such categories. This is clearly due to the huge number of parameters that the model induces: *BH-Forced* estimates the matrix $\{\varphi_{k,i,l}\}_{k=1,\ldots,K;i=1,\ldots,N;l=1,\ldots,L}$ which is one order of magnitude bigger than the same matrix in the other co-clustering models (like *Bi-LDA* or *BH-Free*).

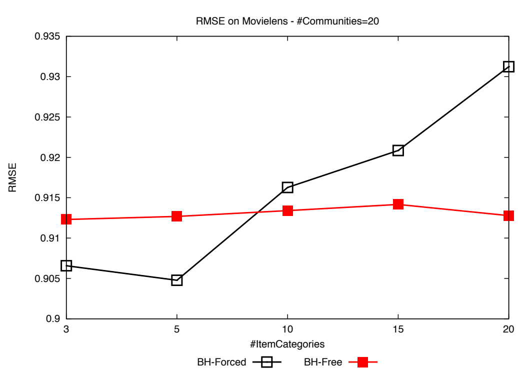


Figure 5: RMSE on MovieLens data - Bayesian Hierarchical Model (#usercommunities=20)

As shown in Fig. 6, the learning time of the *BH* models introduced a reasonable overhead with respect to the learning time of *LDCC*, when the number of item categories is less than 20.

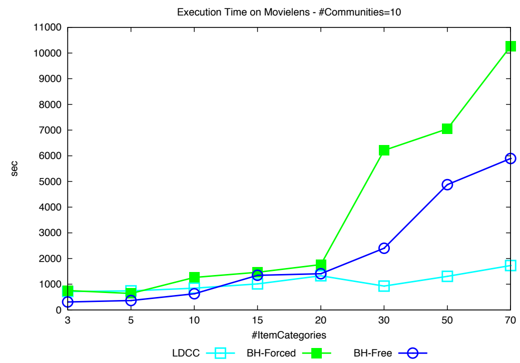


Figure 6: Execution Time on MovieLens data

Recommendation Accuracy. Things change substantially when considering the precision and recall accuracy metrics described in Sec. 2. Based from the results in [2, 4], we consider here also *LDA* model, which has been identified as one of the top-performers in terms of recommendation accuracy. Notice that *LDA* was not included in the analysis of predictive accuracy, as it does not explicitly support a way to compute rating prediction. [2]

The recommendation list for traditional probabilistic approaches based on forced prediction is computed by sorting items according to the expected value. As far as *HUCM* is concerned, even if the overall model does not specify item-selection probabilities, these components are modeled explicitly by the simplified non-hierarchical *(B)UCM* versions (detailed in [2, 4, 5]). To summarize, we equip *LDA* with item selection ranking, and *UCM*, *BUCM* and *BH-Free* with item selection and relevance ranking. All the other approaches are based on the expected value.

Figures 7 and 8 show the results in recommendation accuracy on Movielens and Netflix data, when the size k of the list varies from 1 to 20. Probabilistic models equipped with item-selection achieve the best results in both datasets. On Movielens data, *BH-Free* follows the same trend as *LDA* for user satisfaction, and exhibits a minimal worsening on standard recall (0.39 vs 0.37) and precision (0.11 vs 0.10). *BF-Forced* does not compare with item-selection methods, but achieves competitive results with the remaining probabilistic co-clustering approaches, outperforming them in user satisfaction recall. Notably, the discrepancy between the recommendation accuracy of Bayesian approaches and the non-bayesian ones is consistently large. In particu-

[2] As a matter of fact, extensions explicitly modeling such feature [16] have been experimentally shown in [4] to perform worse than *PMF*.

	MovieLens		Netflix	
Approach	**Best RMSE**	**#Topics**	**Best RMSE**	**#Topics**
PMF	**0.8655**	10	0.9309	100
HUCM	0.9278	2-3	**0.9212**	50-10
Bregman-CC	0.9023	10-20	0.9873	3-5
Bi-LDA	0.9033	30-20	0.9362	30-15
LDCC	0.9074	5-5	0.9419	5-10
BH-Forced	0.9041	15-3	0.9320	10-3
BH-Free	0.9073	30-5	0.9256	30-5
BUCM	0.9292	30	0.9431	10

Table 3: Summary of predictive accuracy over the MovieLens and Netflix datasets

lar, both *BUCM* and **BH-Free** outperform *UCM*. This confirms the advantages of the Bayesian approach.

The trends are confirmed and even strengthened on Netflix data: approaches equipped with item-selection and relevance ranking, and in particular *BH-Free*, tend to outperform all the other approaches. *BH-Free* achieves the best recommendation accuracy and exhibits a global gain over both *UCM* and *BUCM*.

The outperformance of *BUCM* over *BH-Free* in Movielens can be explained by the different distribution of these data with regards to Netflix. In this latter case, in fact, the huge volume of data is more likely to exhibit local patterns, which are better modeled by *BH-Free*. By converse, Movielens exhibits both less users and less ratings, and hence the simpler *BUCM* model can easily fit the data.

5 Final Remarks and Conclusion

In this work we proposed a hierarchical Bayesian approach for preference data, which extends state-of-the-art (hierarchical) co-clustering techniques, by modeling dynamic associations and dependencies between user- and item-clusters. Two versions of the general schema were proposed, namely *BH-Forced* and *BH-Free*, respectively based on the forced- and free-prediction semantics. An extensive evaluation was performed to assess the skills of the devised models, in terms of both rating prediction and recommendation accuracy. *BH-Free* and *BH-Forced* were shown to achieve a competitive prediction accuracy on the Movielens and Netflix data sets, with respect to co-clustering competitors. However, the two models perform differently as the number of item categories grows. In fact, *BH-Forced* tends to overfit, while the incorporation of item selection results in the more robust *BH-Free* model. The learning time of the proposed approaches is comparable to those of other co-clustering techniques with a reasonable overhead due to the higher structural complexity of the proposed models.

BH-Free is characterized by a high recommendation accuracy: on the Movielens data set, it achieves competitive results with respect to *LDA*, and it outperforms all competitors on the sample of the Netflix collection. Table 4 summarizes prediction and recommendation performances on both datasets, reporting for each model the settings that achieve the best results in terms of recommendation accuracy. Due to space limitations, we report only values for US-Recall and US-Precision. This final comparison highlights the effectiveness of the proposed *BH* models, which represent the best compromise between prediction and recommendation accuracy.

We plan to extend the proposed model in two main directions. First of all, we are interested in combining in the same bayesian framework both collaborative and content features. This is expected to increase the accuracy of the recommendations provided by the system and the background content information can be used to provide personalized recommendations in cold-start scenarios. Moreover, since the users behavior on web is always more influenced by its social interactions with the other users, and *social recommender systems* [26, 27] are emerging as a powerful combination of both recommendation and social networking features, we are interested in providing an extension of the proposed framework which takes into account both users' past preferences and explicit people relationships to enhance recommendations.

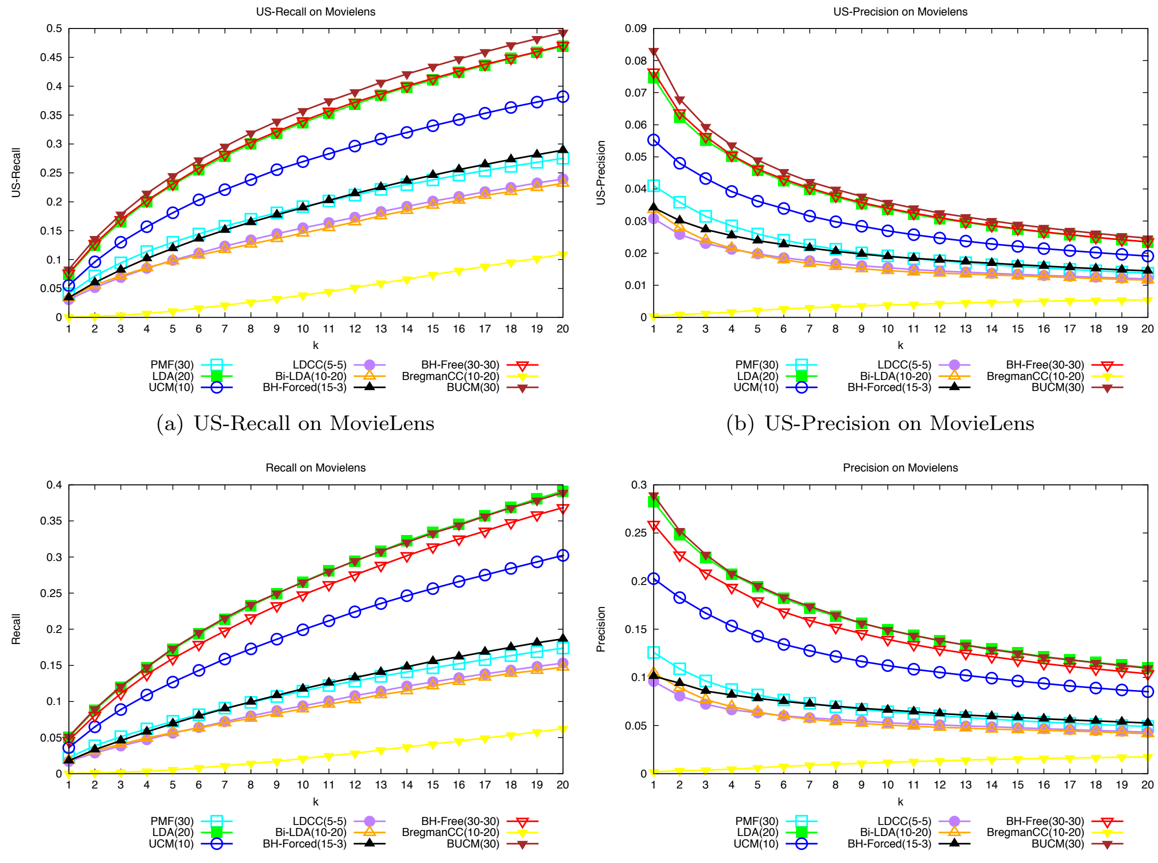
References

[1] N. Barbieri, *Regularized Gibbs Sampling for User Profiling with Soft Constraints*, in Proc. Int. Conf. on Advances in Social Networks Analysis and Mining (ASONAM'11), 2011.

[2] N. Barbieri, G. Costa, G. Manco and R. Ortale, *Modeling Item Selection and Relevance for Accurate Recommendations: A Bayesian Approach*, in Proc. ACM Conf on Recommendation Systems (RecSys'11), 2011.

[3] N. Barbieri, M. Guarascio and G. Manco, *A Block Mixture Model for Pattern Discovery in Preference Data*, in Proc. ICDM Workshop on Topic Feature Discovery and Opinion Mining, 2010.

[4] N. Barbieri and G. Manco, *An Analysis of Probabilistic Methods for Top-N Recommendation in Collabora-*



(a) US-Recall on MovieLens

(b) US-Precision on MovieLens

(c) Recall on MovieLens

(d) Precision on MovieLens

Figure 7: Precision and recall over the MovieLens data set

	MovieLens				Netflix			
Approach	**#Topics**	**RMSE**	**US-Recall**	**US-Precision**	**#Topics**	**RMSE**	**US-Recall**	**US-Precision**
PMF	30	0.8714	0.2752	0.0137	100	0.9309	0.2285	0.0114
LDA	20	-	0.4689	0.0234	20	-	0.5089	0.0254
UCM	10	0.9431	0.3820	0.0190	20	0.9290	0.5390	0.0269
LDCC	5-5	0.9074	0.2394	0.01197	20-15	0.9612	0.2241	0.0112
Bi-LDA	10-20	0.9033	0.2321	0.0116	15-3	0.9433	0.2010	0.010
BH-Forced	15-3	0.9041	0.290	0.0144	10-10	0.9357	0.2468	0.0123
BH-Free	30-30	0.9073	0.4706	0.0235	30-5	0.9256	0.5719	0.0286
Bregman-CC	10-20	0.9023	0.110	0.0054	3-5	0.9873	0.1546	0.0077
BUCM	30	0.9292	0.493	0.0247	10	0.9431	0.5347	0.0267

Table 4: Comparison of Predictive and Recommendation Accuracy on MovieLens and Netflix data

tive Filtering, in Proc. ECML-PKDD Conf., Athens, Greece, 2011.

[5] N. Barbieri, G. Manco and E. Ritacco *A Probabilistic Hierarchical Approach for Pattern Discovery in Collaborative Filtering Data*, in Proc. SDM Conf. , 2011.

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

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

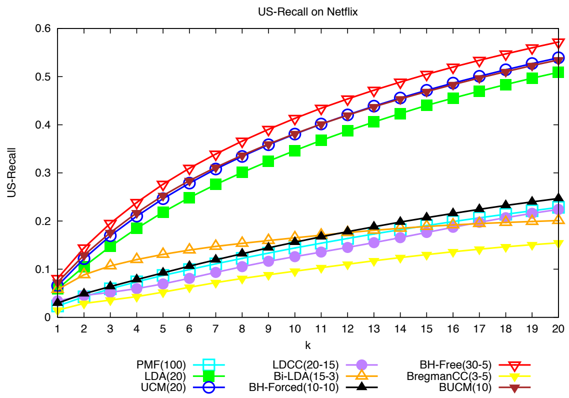
[8] E. Campochiaro, R. Casatta,P. Cremonesi and R. Turrin *Do Metrics Make Recommender Algorithms?*, in AINA Workshops, 2009.

[9] P. Cremonesi, Y. Koren and R. Turrin, *Performance of recommender algorithms on top-n recommendation tasks*, in Proc. ACM Conf on Recommendation Systems (RecSys'10), 2010.

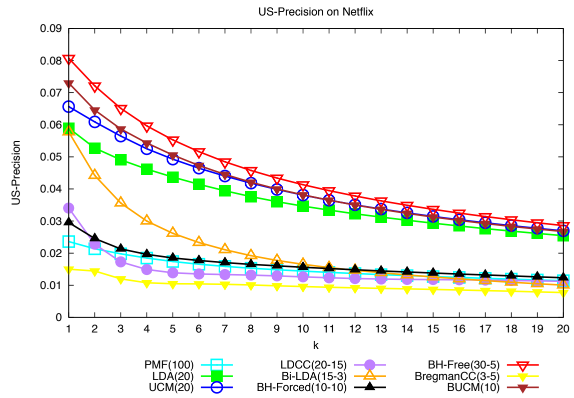
[10] T. George and S. Merugu, *A Scalable Collaborative Filtering Framework Based on Co-Clustering*, in ICDM, 2005.

[11] G. Heinrich, *Parameter Estimation for Text Analysis*, University of Leipzig, `http://www.arbylon.net/publications/text-est.pdf`, 2008.

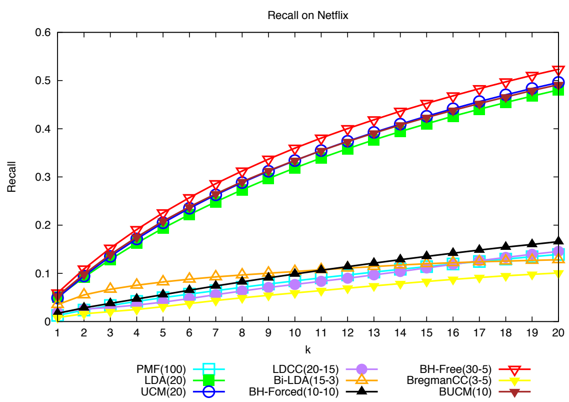
[12] J. L. Herlocker, J. A. Konstan, L. G.Terveen and J. T.



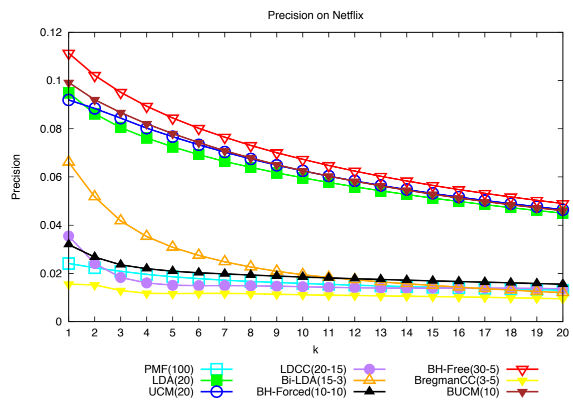
(a) US-Recall on NetFlix



(b) US-Precision on NetFlix



(c) Recall on NetFlix



(d) Precision on NetFlix

Figure 8: Precision and recall over the Netflix data set

Riedl, *Evaluating collaborative filtering recommender systems*, ACM Transactions on Information Systems (TOIS), 22 (2004), pp. 5–53.

[13] T. Hofmann *Collaborative filtering via gaussian probabilistic latent semantic analysis*, in Proc ACM Conf on Information Retrieval (SIGIR'03), 2003.

[14] T. Hofmann, *Latent semantic models for collaborative filtering*, ACM Transactions on Information Systems (TOIS), 22 (2004), pp. 89–115.

[15] T. Hofmann and J. Puzicha, *Latent class models for collaborative filtering*, in Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI'99), 1999.

[16] B. Marlin, *Modeling user rating profiles for collaborative filtering*, in NIPS, 2003.

[17] S. M. McNee, J. Riedl and J. A. Konstan, *Being Accurate is Not Enough: How Accuracy Metrics have hurt Recommender Systems*, in ACM SIGCHI Conference on Human Factors in Computing Systems, 2006.

[18] I. Porteous, E. Bart and M. Welling, *Multi-HDP: a non parametric Bayesian model for tensor factorization*, in AAAI, 2008.

[19] F. Ricci, L. Rokach,B. Shapira and P.B. Kantor *Recommender Systems Handbook*, Springer, 2011

[20] R. Salakhutdinov and A. Mnih, *Probabilistic Matrix Factorization*, in NIPS, 2008.

[21] B. Sarwar, G. Karypis, J. Konstan and J. Reidl, *Item-based collaborative filtering recommendation algorithms*, in WWW, 2001.

[22] H. Shan and A. Banerjee, *Bayesian Co-clustering*, in ICDM, 2008.

[23] L. Si and R. Jin, *Flexible Mixture Model for Collaborative Filtering*, in ICML, 2003.

[24] L. Si, R. Jin and C. Zhai, *A study of mixture models for collaborative filtering*, Information Retrieval, 9 (2006), pp. 357–382.

[25] P. Wang, C. Domeniconi and K. B. Laskey, *Latent Dirichlet Bayesian Co-Clustering*, in ECML-PKDD, 2009.

[26] H. Ma, H. Yang, M.R. Lyu, I. King, *SoRec: social recommendation using probabilistic matrix factorization*, in CIKM,2009

[27] M. Jamali and E. Martin, *A matrix factorization technique with trust propagation for recommendation in social networks*, in Proc. ACM Conf on Recommendation Systems (RecSys'10), 2010