

A Probabilistic Hierarchical Approach for Pattern Discovery in Collaborative Filtering Data

Nicola Barbieri*

Giuseppe Manco[†]

Ettore Ritacco[‡]

Abstract

This paper presents a hierarchical probabilistic approach to collaborative filtering which allows the discovery and analysis of both global patterns (i.e., tendency of some products of being ‘universally appreciated’) and local patterns (tendency of users within a community to express a common preference on the same group of items). We reformulate the collaborative filtering approach as a clustering problem in a high-dimensional setting, and propose a probabilistic approach to model the data. The core of our approach is a co-clustering strategy, arranged in a hierarchical fashion: first, user communities are discovered, and then the information provided by each user community is used to discover topics, grouping items into categories. The resulting probabilistic framework can be used for detecting interesting relationships between users and items within user communities. The experimental evaluation shows that the proposed model achieves a competitive prediction accuracy with respect to the state-of-art collaborative filtering approaches.

1 Introduction

Recommender Systems (RS) provide users with personalized suggestions about new products, services, and information. According to the *collaborative filtering (CF)* approach [1], RS are based exclusively on a database of user preferences. The underlying assumption of *CF* techniques is that users who had shown similar preferences in the past, will tend to make similar choices in the future. Traditional CF approaches [2–5] try to foresee the preferences of users on previously unseen products, by analyzing and discovering patterns in a high dimensional and extremely sparse preference matrix. Recently, generative probabilistic models [6–10] have been gaining increasing attention, as they offer several advan-

tages with respect to non-probabilistic techniques: e.g., they allow the estimation of predictive distributions, the possibility to include prior knowledge into the generative process and a principled framework to select model structure. Moreover, the inferred latent structure is directly tied to the generative process and therefore often easy to interpret.

Collaborative filtering data exhibit global patterns (i.e. tendencies of some products of being ‘universally’ appreciated) as well significative local patterns (i.e tendency of users belonging to a specific community to express similar preference indicators on the same items). Local preferences affect the performance of RS especially when the number of users and items grows, and their importance has been acknowledged by the current CF literature [11].

Typically, local patterns can be better detected by means of co-clustering approaches [12–16]. Unlike traditional CF techniques, which try to discover similarities between users or items using clustering techniques or matrix decomposition methods, co-clustering approaches aim to partition data into homogenous blocks enforcing a simultaneous clustering on both the dimensions of the preference data. This highlights the mutual relationships between users and items: similar users are detected by taking into account their ratings on similar items, which in turn are identified considering the ratings assigned by similar users.

However, a main weakness of the current approaches to co-clustering is the static structure enforced by fixed row/column blocks where both users and items have to fit. For example, the movies “Titanic” and “Avatar”, are typically associated with different categories: the former is about romance, whereas the latter can be considered an action, sci-fi movie. Assuming a global and unique partition on the item-set, we can expect to see the movies into different partitions. However, that structure would fail to recognize a group of users who are really into the movies of James Cameron (the director of both movies). Analogously, any method associating the two movies with the same partition would fail in catching the difference in genre.

The issue in the previous example is that different

*DEIS, University of Calabria, via Bucci 41c, 87036 Rende (CS) - Italy. Email: nbarbieri@deis.unical.it

[†]ICAR-CNR, via Bucci 41c, 87036 Rende (CS) - Italy. Email: manco@icar.cnr.it

[‡]ICAR-CNR, via Bucci 41c, 87036 Rende (CS) - Italy. Email: ritacco@icar.cnr.it

user groups can infer different interpretations of item categories. A more flexible structure, where item categories are conditioned by user categories, would better model such situation, by e.g., allowing “Titanic” and “Avatar” to be observed in the same item category within the “Cameron” group, and in different categories outside. Notice that traditional clustering approaches are not affected by this problem, as they only concentrate on local patterns in one dimension of the rating matrix. The drawback, however, is that they ignore structural information in the other dimension, which by the converse can be exploited both for more accurate prediction and user profiling.

This paper presents a novel probabilistic hierarchical approach which is able to discover both global and local trends in data, allowing different user communities to show different preference values on distinct groups of items. The proposed schema differs from the previously proposed coclustering approaches to CF data because it does not assume the existence of a unique partition on the item-set: each user community is characterized by having its own set of topics involving items and user preferences. Following a hierarchical clustering approach, we initially determine user communities by gathering together similar users. Then, for each user community the clustering phase produces a mixture of topics upon which the item set and the user preferences are accommodated into categories. Each item group is characterized by the *intracluster consistency property* with respect to the considered user community: each item and its neighbors, associated by having received common rating value in the context of the community, will belong to the same cluster with high probability.

The hierarchical coclustering model does not enforce any strong assumption on the membership of users and items improving the flexibility of the model itself. Each user participates to different user communities with a certain degree and, given a user community, each item may belong to different item-categories. As a result, the proposed model summarizes the advantages of a flexible probabilistic structure for user profiling and a competitive prediction accuracy on user ratings.

The rest of the paper is organized as follows. In Sec. 2 we define a formal framework of the collaborative filtering data, and state of art approaches to rating prediction will be briefly discussed in Sec. 3. The hierarchical probabilistic model for CF data is then presented and formalized in Sec. 4 and its performances are analyzed and discussed in Sec. 5.

2 Background

A RS consists of a set of M users $\mathcal{U} = \{u_1, \dots, u_M\}$, which will be indicated for short as the *user-set*, a set

of N items $\mathcal{I} = \{i_1, \dots, i_N\}$, named *item-set*, and a collection of rating values expressing the preference of one user on a corresponding item. Such collection of preference indicators can be represented as a $M \times N$ rating matrix \mathbf{R} , where r_i^u is the rating given by the user u on the item i . Ratings can be integer values within a scale 1 (low interest) to V (strong interest). Even in the case of a very dynamic system, the rating matrix is typically characterized by an exceptional sparsity rate; if the rating for the pair (*user, item*) is unknown we will assume $r_i^u = 0$.

Let $\mathcal{U}(i)$ the set of users who evaluated the item i , while we will denote by $\mathcal{I}(u)$ the set of all the items for which the user u has expressed her preference. An example of rating matrix with $M = 7$ users and $N = 5$ items is shown in Fig 1. The goal of a RS is to learn

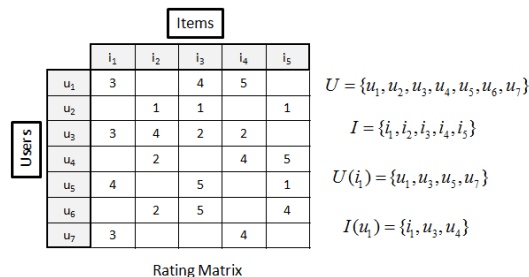


Figure 1: An example of rating matrix

a preference function $p : \mathcal{U} \times \mathcal{I} \rightarrow \{1, \dots, V\}$, which associates to each pair (*user, item*) a rating value within the admissible range. Let \hat{r}_i^u denote the predicted rating for the pair (u, i). Considering the case of users and products which have provided/received at least one preference value, several evaluation metrics have been proposed to quantify the quality of a prediction algorithm. Denoting by \mathcal{T} a test-set collection of triples (*user, item, rating*), one of the most referenced methods to measure the performance of a predictor is the Root Mean Squared Error, which emphasizes large errors:

$$RMSE = \sqrt{\frac{\sum_{(u,i) \in \mathcal{T}} (r_i^u - \hat{r}_i^u)^2}{|\mathcal{T}|}}$$

In a probabilistic settings, we adopt random variables R , I , and U denoting a rating, an item and a user respectively. Then, $P(R = r)$ will denote the probability to observe a rating with value r , and analogously $P(U = u)$ will denote the probability that a given rating has been give by user u . With an abuse of notation, we shall omit the random variable in the specification. For example, $P(r, u, i)$ shall denote the joint probability $P(R = r, U = u, I = i)$.

3 Collaborative Filtering Approaches

In this section we provide a brief discussion of the most used techniques for rating prediction, namely *Baseline*, *Nearest Neighbors* and *Latent Factor* models.

Baseline models are basic techniques to compute a rating prediction and are considered a first step towards the rating personalization and user profiling. These first approaches to rating prediction are summarized in Table 1, where μ denotes the overall mean of the rating, \bar{r}_i is the average rating given on the item i and, symmetrically, \bar{r}_u is the average of the ratings given by the user u . Another simple and effective baseline

Baseline	Personalization	Prediction \hat{r}_i^u
OverallMean	None	μ
ItemAvg	Item-oriented	\bar{r}_i
UserAvg	User-oriented	\bar{r}_u
WeightedCentering	Item&User oriented	$\alpha \bar{r}_i + (1 - \alpha)\bar{r}_u$ $0 \leq \alpha \leq 1$

Table 1: Baseline Approaches

approach has been proposed in [17], which describes a set of global effects that might influence user's ratings. For example, some users might tend to assign higher or lower ratings to items respect to their rating average (which is known as *User effect*), while some items tend to receive higher (or lower) rating values than others (*Item effect*).

Neighborhood based approaches based on explicit user feedback are the most commonly used techniques for generating suggestions and predictions. According to the item-based version [2], the predicted rating is computed by aggregating the ratings given by each user on the most similar items to the considered item. The underlying assumption is that the user might prefer items more similar to the ones she liked before, because they might belong to the same category or might share similar features. More formally:

$$\hat{r}_i^u = \frac{\sum_{j \in \mathcal{N}^K(i;u)} s_{ij} \cdot r_j^u}{\sum_{j \in \mathcal{N}^K(i;u)} s_{ij}}$$

where $\mathcal{N}^K(i;u)$ is the set of K items rated by the user u most similar to i , and s_{ij} are the similarity coefficients between the item i and j . Similarity coefficients are computed on a global basis: two products are considered similar if they have received similar preference values from common users. Hence, this strategy fails in recognizing local item similarity within a same user community. Some alternative and more effective formulations of the neighborhood based approach have been proposed in [5,17,18]; the key idea is to determine the interpolation weights simultaneously according to a global optimization schema, which better reflects intra neighborhood relationships.

The assumption behind *Latent Factor models* is that the rating value can be expressed by considering a set of contributes which represent the interaction between a user and the target item on a set of features. Assuming that there are a set of K features which determine the user's preference on an item, the prediction can be generated as:

$$r_i^u = \sum_{z=1}^K U_{u,z} M_{z,i}$$

where $U_{u,z}$ is the response of the user u to the feature z and $M_{z,i}$ is the response on the same feature by item i . Several learning schemes have been proposed to overcome the sparsity of the original rating matrix and to produce accurate predictions. The learning phase relies either on a *gradient descent* error-minimization [3,4] or a likelihood optimization procedure (based e.g., on Gibbs Sampling or Expectation Maximization). The peculiarity of a probabilistic model is the capability of estimating either the joint probability $P(r, u, i)$ or the conditional probability $P(R = r|u, i)$. The identification of \hat{r}_i^u for a pair $\langle u, i \rangle$ can hence be computed as:

$$\hat{r}_i^u = E[r|u, i] = \sum_r r \cdot P(R = r|u, i)$$

The *pLSA* (probabilistic latent semantic analysis, or *Aspect Model*) proposed by Hoffman in [13], is the reference probabilistic approach to CF. The underlying assumption is that the observed user preferences can be modeled as a mixture of user communities, and each user can be included into one or more groups [13]. Introducing a latent variable Z (ranging over K possible states) and assuming that user U and item I are conditionally independent given the state of Z , the probability of observing rating r for the pair (u, i) can be computed as:

$$P(r|u, i) = \sum_{z=1}^K P(r|i, z)P(z|u)$$

where $P(z|u)$ represents the interest of u to topic z , and $P(r|i, z)$ is the probability that a user belonging to pattern z gives rating r on the item i .

The *User Profile Model* extends this formulation by employing Dirichlet priors which provide a full generative model at the user level [19,20]; a further adoption of the Latent Dirichlet Allocation approach [21] which includes a response variable, in this case the ratings on an item, has been proposed in [22].

Recently, novel probabilistic approaches [6,23,24] have been proposed to overcome the need for regularization and in order to prevent overfitting in matrix factorization methods. In particular, the *Probabilistic Matrix*

Factorization [6] proposes a generative gaussian model for ratings, in the low-rank latent space of users and items. Extensions of this model include bayesian priors [8] and non-linear matrix factorization with gaussian processes [25].

Other works focus on combining preference data and content features [7, 9] to produce more accurate recommendations and to address the cold-start problem. The underlying idea is to associate items and users with content-specific latent factors and thus to use this low-dimensional feature representations for regularization.

So far, co-clustering approaches exhibited limited predictive capability (clustering both items and users makes these approaches more prone to overfitting). In addition, the high computational burden make them unfeasible for realistic problems. [12] proposes simultaneous clustering of users and items based on an adaptation of the *Bregman coclustering* [26]: given an initial co-clustering assignment, the user-clusters (rows) and item-clusters (columns) are alternately optimized till convergence is achieved. A probabilistic approach to determine user-item memberships following a coclustering strategy has been discussed in [13]: the assumption behind the *Two-Sided Clustering Model* is that the rating value is independent of the user and item identities given their respective cluster memberships. The clustering approach is based on a standard EM likelihood maximization procedure.

Both these co-clustering approaches assume the existence of a unique partition over the item-set and the number of user-communities. Within these models, each user belongs to exactly a single user-community and each item belong to a single groups of item. By contrast, the *Flexible Mixture Model (FMM)* [14] extends the two-sided model by allowing an individual (either a user or an item) to be included in different clusters, with different degrees of membership. A novel approach to co-clustering have been proposed in [10]; the resulting model, known as *Bi-LDA*, integrates Dirichlet priors and discovers simultaneous groups of users/items modeled via LDA.

4 A Hierarchical Co-Clustering Approach for Modeling User Preferences

The starting point in our approach is the observation that different communities can infer different evaluations of the same item. Specific groups of users tend to be co-related according to different subsets of features. However, though semantically-related, two users with (possibly several) differences in their item ratings would hardly be recognized as actually similar by any global model imposing a fixed structure for item categories. Individual user can be intended as a mixture of latent con-

cepts, each of which being a suitable collection of characterizing features. Accordingly, two users are considered as actually similar if both represent at least a same concept. Viewed in this perspective, the identification of *local patterns*, i.e. of proper combinations of users and items, would lead to the discovery of natural clusters in the data, without incurring into the foresaid difficulties. Consider the toy example in Fig. 2, where homogenous 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.

		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		$d_3 = \{i_8, i_9\}$
	u_4		1	1		5	4	5	4		2	
Community 2	u_5	5		4	5	5	1	4	3		1	$d_4 = \{i_1, i_4, i_5\}$
	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 2: Example of Local Pattern in CF Data

The generative model for the proposed scheme can be summarized as follows:

1. Select a user community c_k according to the probability distribution π_k ;
2. select a user u with probability $P_k(u) = P(u|c_k)$ and an item i with probability $P_k(i) = P(i|c_k)$;
3. Choose a topic d_h with probability $P(d_h|i, c_k)P(d_h|u, c_k)$;
4. produce the rating r with probability $\phi_h(r) = P(r|d_h)$.

Formally, we can assume that the probability of a triplet $\langle u, i, r \rangle$ is

$$P(u, i, r) = \sum_{k=1}^K \pi_k P_k(u) P_k(i) P(r|i, u, c_k) \quad (4.1)$$

where

$$P(r|i, u, c_k) = \sum_{h=1}^{H_k} \phi_h(r) P_k(d_h|i) P_k(d_h|u) \quad (4.2)$$

The latter correspond to a “local” probabilistic latent semantic analysis, provided that the user communities are known.

The idea, in the above formula, is learning latent communities from the data as well as a collection of characterizing concepts for each community. In particular, each rating can be seen as the outcome of a mixture of various concepts, where some concepts are more or less probable according to the cluster where the user fits. Hence, a data tuple can be thought as the outcome of the following generative model: firstly pick a distribution over latent clusters; next, choose the concepts associated and finally generate the individual values. Also, notice the role of the $\pi_k, k = 1, \dots, K$ prior probabilities in the generative process. In practice, they model the assumption that observing a pair $\langle u, i \rangle$ is not totally random, but it is instead the result of the grouping of users into communities.

Due to the strong coupling between the user community latent variable c and the one corresponding to local patterns d , the exact inference for the model characterized by the joint probability in Eq. 4.1, which would maximize both the user community cohesion and the local topic similarity, is difficult to solve analytically. Hence, we adopt an approximated solution, based on a hard clustering policy for user communities, such that the inference of the parameters can be performed efficiently without compromising the generative semantic and the flexibility of the model.

We devise a hierarchical approach to the estimation of the components involved into Eq. 4.1. In practice, our approach consists in a preliminary discovery structure, where user communities are detected. Next, for each user community, a topic model is investigated, and the most prominent topics are discovered and properly modeled.

The general scheme of the algorithm is shown in Alg. 1 and could be summarized as follows: given a rating matrix \mathbf{R} , discover k user communities; then, for each of those communities, according to an hard clustering approach, select from \mathcal{U} a subset of users that belong to the considered community and generate a set of H_k topic models for their ratings.

The hierarchical model for users’ ratings consists in a set of K user community models and for each of them a set of H_k topic models which represent local preference patterns for the member of the considered community. The user community level specifies the probabilities $\gamma_{uk} = P(c_k|u)$ with $k = 1, \dots, K$, which measure how much the ratings given by the user u fit the preference behaviour underlined by each of the communities.

The probability of observing the rating r for the

Algorithm 1 HMbuild

Require: The sets $\mathcal{U} = \{u_1, \dots, u_M\}$ and $\mathcal{I} = \{i_1, \dots, i_N\}$ and the corresponding rating matrix \mathbf{R} ;

Ensure: a set $\mathcal{C} = \{c_1, \dots, c_K\}$ of user community models and a subset $\mathcal{D}^k = \{d_1^{(k)}, \dots, d_{H_k}^{(k)}\}$ for each user community k

- 1: $\mathcal{C} \leftarrow \text{GenerateUserCommunities}(\mathbf{R})$;
 - 2: **for all** community model $c_k, k = 1, \dots, K$ **do**
 - 3: let $\mathcal{U}_k = \{u \in \mathcal{U} | p(c_k|u) \geq p(c_j|u), j = 1, \dots, K\}$, and \mathbf{R}_k the corresponding submatrix of \mathbf{R} ;
 - 4: $\mathcal{D}^k \leftarrow \text{GenerateTopicModels}(\mathbf{R}_k)$;
 - 5: **end for**
-

pair (u, i) can be computed considering two schema, summarized in Alg. 2:

- *Hard-Clustering Prediction:*

$$P(r|i, u) = \sum_{h=1}^{H_k} \phi_h(r) P_k(d_h|i) P_k(d_h|u) \quad (4.3)$$

where $k = \operatorname{argmax}_{j=1, \dots, K} \gamma_{uj}$ is the cluster that better represents the previously observed rating of the user u . This prediction rule relies exclusively on the information given by the topic model corresponding to the user’s cluster; thus it might produce low quality predictions if the user’s community is not identified with enough confidence.

- *Soft-Clustering Prediction:*

$$P(r|i, u) = \sum_k \gamma_{uk} \cdot \tilde{P}(r|i, u, c_k) \quad (4.4)$$

where the probabilities γ_{uk} act as mixture weights and the distribution over rating values corresponding to the community c_k is computed taking into account both global and local patterns:

$$\tilde{P}(r|i, u, c_k) = \begin{cases} P(r|i, u, c_k) & \text{if } u \in \mathcal{U}_k \\ P(r|i, c_k) & \text{otherwise} \end{cases} \quad (4.5)$$

Note that if $u \in \mathcal{U}_k$ then γ_{uk} is the dominant mixing weight and the distribution over ratings is refined by considering the corresponding set of topic models; in the opposite case the distribution over ratings can be estimated by considering the probability of observing each rating given an item within the considered community.

4.1 Modeling User Communities. The discovery of the communities is accomplished essentially via a model-fitting procedure based on a maximum-likelihood estimation. In practice, we assume that the rating

Algorithm 2 HMcomputeRatingsProbability

Require: a pair $\langle u, i \rangle$
Ensure: a probability $P(R = r|u, i)$ for each rating value r

```

1: let  $c = \operatorname{argmax}_{j=1, \dots, K} p(c_j|u)$ 
2: for all  $r = 1$  to  $V$  do
3:   if Hard-Clustering then
4:      $P(R = r|u, i) = \sum_{h=1}^{H_k} \phi_h(r) P_k(d_h|i) P_k(d_h|u)$ 
5:   else
6:     for all community model  $c_k, k = 1, \dots, K$  do
7:       if  $k = c$  then
8:          $prob \leftarrow \mathcal{D}^k.getRatingProbability(r, u, i)$ 
9:       else
10:         $prob \leftarrow c_k.getRatingProbability(r, i)$ 
11:      end if
12:       $P(R = r|u, i) \leftarrow P(R = r|u, i) + \gamma_{uk} \times prob$ 
13:    end for
14:  end if
15: end for

```

matrix \mathbf{R} is modelled as a set of user vectors, where each vector is characterized by the preferences of the user. Formally, this means that we can model the probability $p(r, i|u)$ for each triplet $\langle r, i, u \rangle$.

The corresponding probability of observing a user hence corresponds to the joint probability of observing all his ratings, that is

$$P(u|\Theta, \mathbf{R}) = \prod_{i=1}^N \prod_{r=1}^V (P(i|\Theta) \cdot P(r|i, \Theta))^{\delta(u, i, r)}$$

where

$$\delta(u, i, r) = \begin{cases} 1 & \text{if } r_i^u = r \\ 0 & \text{otherwise} \end{cases}$$

This modeling allows us to adopt a maximum likelihood approach to the estimation of the Θ parameters characterizing the $P(i|\Theta)$ and $P(r|i, \Theta)$. For example, we can characterize $P(i|\Theta)$ via a bernoullian pdf parameterized by α_i , and $P(r|i, \Theta)$ as a multinomial (with factors σ_{ri}). Within a ML framework, the estimation of the above probabilities would produce

$$\alpha_i = \frac{\sum_{u=1}^M \sum_{r=1}^V \delta(u, i, r)}{\sum_{u=1}^M \sum_{i'=1}^N \sum_{r=1}^V \delta(u, i', r)} \quad (4.6)$$

$$\sigma_{ri} = \frac{\sum_{u=1}^M \delta(u, i, r)}{\sum_{u=1}^M \sum_{r'=1}^V \delta(u, i, r')} \quad (4.7)$$

A first effect of the above estimates is to adjust the soft-clustering prediction formula Eq. 4.5 as

$$P(r|i, u) = \beta \sum_k \gamma_{uk} \cdot P(r|u, i, c_k) + (1 - \beta) \sigma_{ri}$$

where β is a weighting factor proportional to the number of ratings $|\mathcal{I}(u)|$. In practice, the estimate σ_{ri} provides a higher contribution when the number of ratings given by a user is low (and hence it acts as a prior).

The component $P(r, u, i, c_k)$ and the posteriors γ_{uk} can be estimated by assuming the existence of a set of communities, where each community models specific user attitudes. In particular, the probability of observing a user is given by the mixture

$$P(u|\mathcal{C}) = \sum_{j=1}^K P(u|c_j) \pi_j = \sum_{j=1}^K \prod_{i=1}^N \prod_{r=1}^V \pi_j (\alpha_{ij} \cdot \sigma_{rij})^{\delta(u, i, r)}$$

where a single community c_j is characterized by the parameters $\alpha_{ij} = P(i|c_j)$ and $\sigma_{rij} = P(r|c_j, i)$. The expected loglikelihood can hence be defined as:

$$\mathcal{Q}(\mathbf{R}; \Gamma) = \sum_{u=1}^M \sum_{j=1}^K \gamma_{uj} \cdot \left[\sum_{i=1}^N \sum_{r=1}^V \delta(u, i, r) \cdot (\log \alpha_{ij} + \log \sigma_{rij}) + \log \pi_j \right]$$

Estimating the parameters by means of an EM procedure yields the following equations:

E-Step:

$$\gamma_{uj} = P(c_j|u) = \frac{P(u|c_j) \cdot \pi_j}{\sum_{j'=1}^K P(u|c_{j'}) \cdot \pi_{j'}}$$

M-Step:

$$\begin{aligned} \pi_j &= \frac{\sum_{u=1}^M \gamma_{uj}}{M} \\ \alpha_{ij} &= \frac{\sum_{u=1}^M \gamma_{uj} \sum_{r=1}^V \delta(u, i, r)}{\sum_{u=1}^M \gamma_{uj} \sum_{i'=1}^N \sum_{r=1}^V \delta(u, i', r)} \\ \sigma_{rij} &= \frac{\sum_{u=1}^M \gamma_{uj} \cdot \delta(u, i, r)}{\sum_{u=1}^M \sum_{r'=1}^V \gamma_{uj} \cdot \delta(u, i, r')} \end{aligned}$$

A further advantage of the above formalization is the possibility of exploiting the above model for prediction purposes as well as for structure discovery. A prediction function in fact can be defined as

$$\hat{r}_i^u = E[R|u, i] = \sum_{r=1}^V r \cdot \sum_k \sigma_{rik} \cdot \gamma_{uk} \quad (4.8)$$

and used as a baseline for the special case described in step 10 of algorithm 2. We shall see in the following

that the resulting baseline function is even competitive with state-of-the art approaches.

The above formalization also allows an alternative gaussian model

$$P(r|i, c_j) = N(v_u^r; \mu_{ij}, \sigma_{ij}) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} \exp\left[-\frac{(v_u^r - \mu_{ij})^2}{2\sigma_{ij}^2}\right]$$

where v_u^r is the Z-score normalization of r with regards to user u :

$$v_u^r = \frac{r - \mu_u}{\sigma_u}$$

and the means and the variances are estimated as proposed in [27].

The rating prediction for the pair (u, i) can be hence computed as:

$$\hat{r}_i^u = \mu_u + \sigma_u \left(\sum_{k=1}^K \gamma_{uk} \cdot \mu_{ik} \right) \quad (4.9)$$

and the **M-Steps** can be rewritten as:

$$\begin{aligned} \mu_{ik} &= \frac{\sum_u \sum_r \gamma_{uk} \cdot \delta(u, i, r) \cdot v_u^r}{\sum_u \sum_r \gamma_{uk} \delta(u, i, r)} \\ \sigma_{ik}^2 &= \frac{\sum_{u=1}^M \sum_r \gamma_{uk} \cdot \delta(u, i, r) (v_u^r - \mu_{ik})^2}{\sum_{u=1}^M \sum_r \gamma_{uk} \delta(u, i, r)} \end{aligned}$$

4.2 Local Community Patterns via Topic Analysis. The approach to the discovery of local community patterns is based again on a EM procedure which aims at maximizing the likelihood of the $\mathbf{R}_k = \{\langle r, u, i \rangle | p(c_k|u) \geq p(c_j|u), j = 1, \dots, K\}$ rating matrix associated to a community model c_k . In practice, we can define the expected log-likelihood

$$\begin{aligned} \mathcal{Q}(\mathbf{R}_k; \Psi) &= \sum_u \sum_i \sum_r \sum_h \psi_k(h; r, i, u) \cdot \\ &[\log \phi_h(r) + \log P_k(d_h|i) + \log P_k(d_h|u)] \end{aligned}$$

where $\psi_k(h; r, i, u) = P(d_h|r, i, u, c_k)$. The EM algorithm can hence be defined in terms of the following formulas:

• **E-Step:**

$$\psi_k(h; r, i, u) = \frac{\phi_h(r) P_k(d_h|i) P_k(d_h|u)}{\sum_j \phi_j(r) P_k(d_j|i) P_k(d_j|u)}$$

• **M-Steps:**

$$\begin{aligned} P_k(d_h|i) &= \frac{\sum_u \sum_r \psi_k(h; r, i, u)}{\sum_{h'} \sum_u \sum_r \psi_k(h'; r, i, u)} \\ P_k(d_h|u) &= \frac{\sum_i \sum_r \psi_k(h; r, i, u)}{\sum_{h'} \sum_i \sum_r \psi_k(h'; r, i, u)} \\ P(r|d_h) &= N(r; \mu_{d_h}, \sigma_{d_h}) \end{aligned}$$

where

$$\begin{aligned} \mu_{d_h} &= \frac{\sum_u \sum_i \sum_r \psi_k(h; r, i, u) \delta(u, i, r) \cdot r}{\sum_u \sum_i \sum_r \psi_k(h; r, i, u) \delta(u, i, r)} \\ \sigma_{d_h} &= \frac{\sum_u \sum_i \sum_r \psi_k(h; r, i, u) \delta(u, i, r) \cdot (r - \mu_{d_h})^2}{\sum_u \sum_r \sum_i \psi_k(h; r, i, u) \delta(u, i, r)} \end{aligned}$$

4.3 Computational aspects. Once the parameters of the hierarchical model have been estimated, the on-line complexity for computing predictions scales with the number of user communities and corresponding topics, while the off-line phase requires more resources. In fact, the complexity of the learning phase is determined by the complexity of discovering user communities, which is linear with the number of observed ratings.

To avoid overfitting, which could deteriorate the predictive skills of the models on unobserved data we adopt an *Early Stopping* criterion: a fraction of the data has been retained as held-out dataset and the models have been trained on the remaining part of the data until the accuracy on the held-out data begins to increase.

The estimation of the correct number of clusters is accomplished by resorting to a Cross-Validation approach based on a penalized Log-Likelihood principle, as described below. Given a set D of observations (in our case, the rating matrix \mathbf{R} and its subsets \mathbf{R}_k), we aim at finding the model parameters Θ maximizing the probability $P(\Theta|D)$. In logarithmic terms,

$$\begin{aligned} \log(P(\Theta|D)) &= \log P(D|\Theta) + \log P(\Theta) \\ &= \log(\mathcal{L}(\Theta|D)) + \log P(\Theta) \end{aligned}$$

The idea in the above formula is to counterbalance two opposing requirements: the fitting of the data and the complexity of the model. By modeling $P(\Theta)$ can be modeled as an exponential distribution w.r.t the size of Θ , we can rewrite the above as

$$\log(P(\Theta|D)) \approx \log(\mathcal{L}(\Theta|D)) - m \log n$$

where m is the size of Θ (i.e., the number of model parameters), and n is the size of D . The evaluation

strategy hence consists in computing $\log(P(\Theta|D))$ for each possible Θ , and in choosing the model where it is maximal. In particular, the strategy can be summarized as follows:

1. fix the values K_{min} and K_{max} ;
2. choose the number C of cross-validation trials;
3. for each trial c :
 - (a) sample a subset D_{train} from D ;
 - (b) for k ranging from K_{min} and K_{max} :
 - (c) compute $\log(P(\Theta_k|D_{train}))^c$;
4. for each K , average the values $\log(P(\Theta_k|D_{train}))^c$ over c ;
5. choose the value k^* such that $\log(P(\Theta_{k^*}|D_{train}))^{avg}$ is maximal.

4.4 Discussion. There are several major differences between the models described in Sec. 3 and the above formalization. Considering pLSA, the hidden variable z there is used to discover similar trends in the rating behavior and encourages grouping users into user communities. The prediction relies solely on $P(r|i, z)$ and does not consider item hierarchies and, hence boosted predictions triggered by similar items. By contrast, the proposed hierarchical approach aims to discover local patterns for each user community. Also, there are two further components which boost the prediction accuracy of the underlying user community model. First, the multinomial prior π_j for each user community j , which helps in preventing overfitting by counterbalancing the contribute of each user u in γ_{uj} . The π_j component can be interpreted as a laplacian smoothing based on uniform Dirichlet priors. Clearly, explicit modeling of such priors via Bayesian estimation, in the style of [19], can be adopted. However, as discussed in the next section, the computational cost would leverage significantly. Also, the α_{ij} component explicitly models the likelihood that item i has been rated within community j . The latter also is a major difference, at the user community level, with respect to the multinomial mixture and the User Rating Profile models, discussed in [20].

Also, notice that the co-clustering techniques discussed in the previous section, like the Flexible Mixture Model, assume the existence of a fixed partition both for user communities and for item categories. In our case instead, each user community is characterized by its own partition over the item-set with a flexible number of topics. In addition, co-clustering models only produce prediction on the basis of local contribution

$P(r|c_k, d_h)$. By contrast, according to Eq. 4.5, our prediction benefits from both local and global information.

A final remark is concerned with the possibility of considering the proposed approach symmetrical. Our model starts with user communities and then generates topics. In theory a dual scheme could be viable as well, by first generating item categories and then specific user communities conditioned to item categories. However, duality only holds if the number of rows and columns of the rating matrix are of the same order of magnitude. In fact, the number of model parameters in an item-based mixture grows linearly to the number of users. If the number of items is significantly less than the number of users, this would cause the generation of few categories characterized by too many parameters (and as a consequence the resulting model would be prone to overfitting).

5 Evaluation

We evaluate the effectiveness of the proposed approach in two different respects:

- To measure the effectiveness of the User community model adopted in the first stage in discovering communities fitting the training data. Since each community should be able to model a user's preferences, it is interesting to measure the prediction accuracy of Eq. 4.8 and Eq. 4.9, which exploit the community mixtures.
- To measure the overall prediction accuracy of the hierarchical approach, and to compare it to other well-known approaches in the literature.

Additionally, as a paradigmatic example, we shall inspect the informative content of the structures discovered by the hierarchical model proposed so far. We will show how the resulting community/topic model can provide significant informative content about the communities and the relevant topics discovered.

We used two popular benchmark datasets (Netflix and Movielens) for rating prediction to validate the predictive performance of the proposed approach. In short, Netflix dataset contains over 100 million of ratings given by 480,189 users on a set of 17,770 movies, collected between October 1998 and December 2005. The Netflix Prize dataset has been the reference data for empirical comparisons of Collaborative Filtering algorithms during the last years, mainly for 3 reasons: (i) size of dataset and sparsness coefficient; (ii) availability of results from competitive algorithms; (iii) availability of a baseline score for the prediction error, achieved by a real RS (the Netflix Cinematch algorithm) on the same dataset. We exploited a subsample of the whole Netflix

	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	

Table 2: Summary of the Data used for validation.

data, and partition it into training and test set, where the latter contains ratings given by a subset of the users in the training set over the same set of items. Info about this dataset are summarized in Table 2.

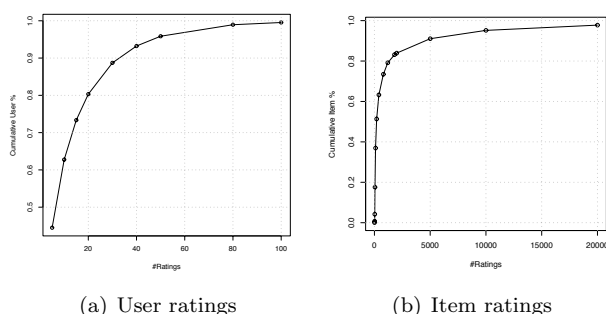


Figure 3: ECDF for user and item ratings on NetFlix.

Fig. 3 shows the empirical cumulative densities for both user and item ratings within the subsample adopted here. There are some major differences between the original Netflix dataset and the subsample used here. For example, we can see from Fig. 3(a) that over 60% of the users have less than 10 ratings and the average number of evaluations given by users is 13 (whereas the original dataset exhibits an average 200 ratings). In addition, figure Fig. 3(b) shows that over 50% of the items have received less than 200 ratings, with an average value of 1929. Again, the average ratings in the original dataset were 5000. In practice, the subsample we exploit is more difficult than the original dataset.¹

The MovieLens-1M² dataset consists of 1,000,209 ratings given by 6,040 users on approximately 3,706 movies; each user in this dataset has at least 20 ratings. We randomly partition the original data into 80% training and 20% test set. Again, Table 2 summarizes

¹This also explains the difference between the values declared in the original papers by the competitors and the values we were able to reproduce on our subsample.

²<http://www.grouplens.org/system/files/ml-data-10M100K.tar.gz>

the values exhibited by the subsets. MovieLens has been a reference dataset for several CF algorithms.

5.1 Predictive Accuracy. We compare our approach with most algorithms mentioned in Sec. 3. In particular, we directly implemented the Regularized SVD [4], pLSA [27], FMM [14], Multinomial Mixture Model [20] and URP [19]. The summary of our results can be found in tables 3(a) and 3(b). Other algorithms not listed here will be discussed separately in the end of the section.

In a first set of experiments, we evaluate the performance achieved by the User Community Models, considering both the Multinomial and the Gaussian version and performed a suite of experiments varying the number of user communities and compared the obtained RMSE values with the ones achieved by the Gaussian pLSA algorithm on the same data.

Experiments on the three models were performed by retaining the 10% of the training (user,item,rating) triplets as held-out data; finally 10 attempts have been executed to determine the best initial configurations. Predictions for the User Community Models are generated according to Eq. 4.5, because preliminary experiments have shown that it outperforms the Hard-Clustering prediction rule. Performance results of the two User Communities Models and pLSA are shown in Figures 4(a) and 4(b).

Considering Netflix, the multinomial User Community approach and the pLSA do not find a significant improvement over the Cinematch base, which is close to 0.95; for both these models the best RMSE values is achieved by considering 150 user communities. The average RMSE for the pLSA model is 0.9474 and only minor improvements on this result are observed varying the number of clusters. The gaussian User Community version outperforms both the multinomial model and pLSA, achieving the best RMSE value of 0.9280 when 30 user communities are employed. The learning phase corresponding to the best model takes about 30 minutes on a INTEL XEON E5520 at 2.27 Ghz, with an average of 6 iterations needed to reach convergence. We were not able to extensively report on FMM and URP on Netflix, due essentially to the high computational resources needed by these models. Table 3(a) shows the best result we were able to compute for FMM. For URP, the only model we were able to compute required 17,340secs and did not exhibit significant results. We were able, however, to thoroughly experiment on MovieLens with these models as well.

Surprisingly, the multinomial User Community model has a significant worsening on MovieLens. While the Gaussian model is still competitive, due essentially

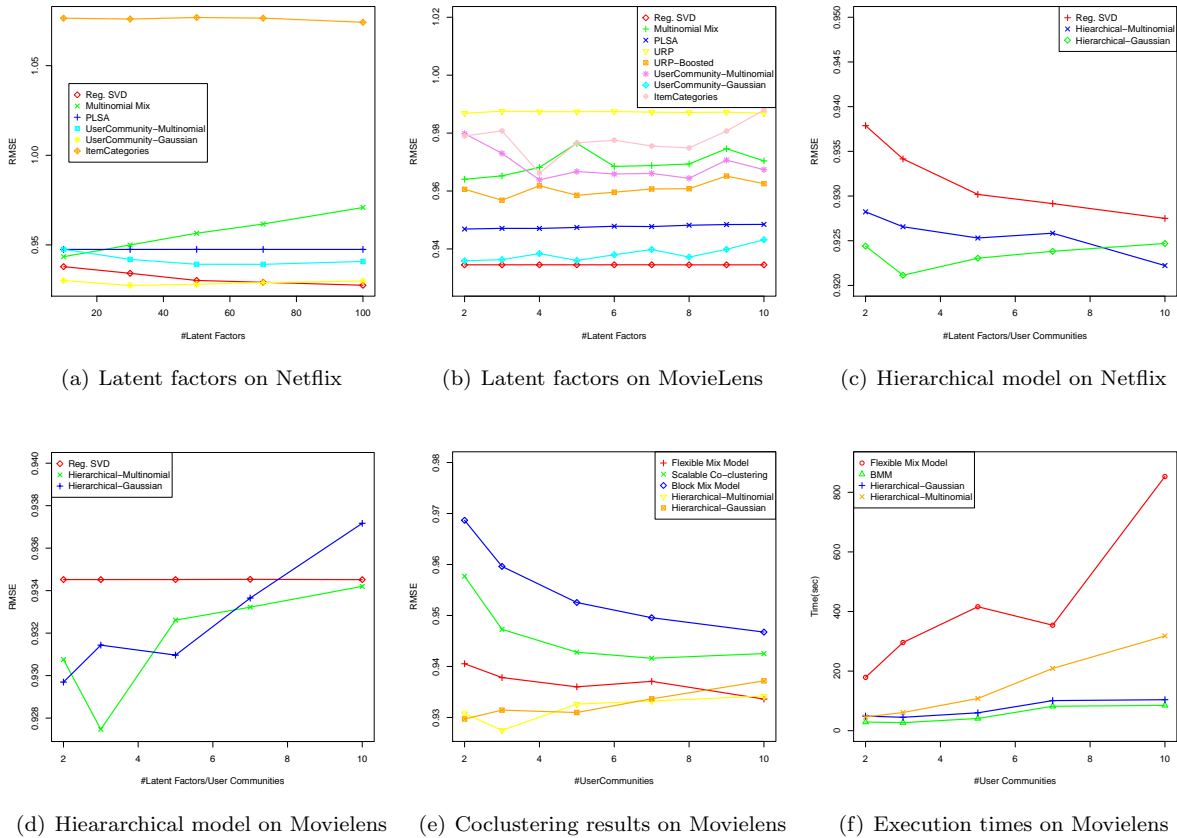


Figure 4: Performance results.

to the z-score normalization exploited in Eq. 4.9, the multinomial model seems to suffer more the skewness of the dataset.

The hierarchical schema allows us to obtain more refined results. This approach has been evaluated by considering both the multinomial and the gaussian version on the first layer clustering, and adopting the procedure for the dynamic estimation of the number of topics described in Sec. 4.3. Fig. 4(d) and Fig. 4(c) show the performances achieved by the two version and the ones achieved by a natural competitor based on latent factors: the regularized SVD. In both the cases, the hierarchical approach produces a significant improvement over the first clustering layer, outperforming the SVD model. On Netflix, hierarchical approach produces RMSE values 0.9222 (multinomial model) and 0.9211 (gaussian model), while the best result achieved by the SVD model is 0.9275. This situation is also reflected in MovieLens where the Reg. SVD produced 0.9345. Again, it's a surprise to see that in this case the multinomial hierarchical approach (0.9274) outperforms the Gaussian hierarchical (0.9296). This result is

even more surprising, if we consider that the multinomial user communities didn't perform very well in the first level. It seems that the adoption of specific item categories boosts the performance significantly.

Figure 4(e) compares all the probabilistic approaches to co-clustering on MovieLens data. Here we compare our approaches with FMM, Bregman Co-clustering [12] and Block Mixture Model [16]. Again, the hierarchical approaches outperform the other co-clustering approaches. This gives evidence that conditioning item categories to user communities provides better structures. Finally, Fig. 4(f) shows the execution times of these co-clustering approaches. Here, we employ 10 item categories and vary the number of user communities.

A final validation qualitatively compares our approach with some among the most popular and effective approaches for making recommendations. We focused on single techniques rather than ensembles or combinations of multiple predictors. Also, we did not take into account models directly modeling temporal aspects, such as the *Time Effect* normalization [17] or the *SVD*

(a) NextFlix Data

Approaches	Best RMSE	Parameters
Overall Mean	1.0839	
User Avg	1.0368	
Item Avg	1.009	
Knn Simple	1.0066	$K = 15$
Scalable Coclustering	0.9862	$K = 3, H = 5$
Weighted Centering	0.9707	$\alpha = 0.6$
Knn with Double Cen. Baseline	0.9637	$K = 20$
Flexible Mixture Model	0.9540	$K = 10, H = 70$
Block Mixture model	0.9477	$K = 30, H = 30$
PLSA	0.9474	$K = 100$
Knn with user effect baseline	0.9453	$K = 20$
Multinomial Mixture Model	0.9434	$K = 10$
User Communities Multinomial	0.9391	$K = 70$
Regularized SVD	0.9275	$\#features = 100$
User Communities Gaussian	0.9274	$K = 30$
KNN Relationship model	0.9258	$K = 20$
Hierarchical model Multinomial - Fixed	0.9251	$K = 50, H = 100$
Hierarchical model Multinomial - Flexible	0.9222	$K = 100$
Hierarchical Model Gaussian - Fixed	0.9212	$K = 50, H = 100$
Hierarchical Model Gaussian - Flexible	0.9211	$K = 30$

(b) MovieLens-1M Data

Approaches	Best RMSE	Parameters
Overall Mean	1.1150	
User Avg	1.0462	
URP	0.9869	$K = 10$
Item Avg	0.9862	
User Communities Multinomial	0.9638	$K = 4$
Multinomial Mix	0.9640	$K = 2$
Weighted Centering	0.961	$\alpha = 0.7$
URP - Boosted	0.9568	$K = 3$
PLSA	0.9468	$K = 2$
Regularized SVD	0.9345	$\#features = 8$
Block Mixture model	0.9467	$K = 10, H = 7$
Scalable Coclustering	0.9416	$K = 7, H = 5$
User Communities Gaussian	0.9359	$K = 2$
Flexible Mixture Model	0.9335	$K = 10, H = 10$
Hierarchical Model Gaussian - Fixed	0.9297	$K = 2, H = 2$
Hierarchical Model Gaussian - Flexible	0.9296	$K = 2$
Hierarchical model Multinomial - Fixed	0.9278	$K = 2, H = 3$
Hierarchical model Multinomial - Flexible	0.9274	$K = 3$

Table 3: Summary of the comparative analysis (K and H represent respectively the number of user communities and item topics)

with temporal dynamics [28]: in fact they exploit temporal information about the preference of the users in a given period in order to refine predictions in the same period, while in a typical setting a recommender system is asked to make suggestions for the future.

Results on Netflix data show that the prediction accuracy achieved by the proposed model is competitive to the ones achieved by other popular recent approaches, such as *PMF* [6], *Bi-LDA* [10] and *SVD++* [5]: the first one is reported to achieve on a sample of 1M ratings of Netflix data an RMSE equals to 0.9253; the latter achieves 0.9333 on the overall Netflix dataset. As far as the *SVD++* is concerned, although it achieves a 0.904 RMSE value on the considered dataset, the problem with such an approach is that it takes advantage of implicit information contained in the test-set.

The model also compares with *fLDA* [7] and the *Regression-based latent factor models* [23], which integrate user/item features and on a 75% – 25% split of the MovieLens-1M achieve 0.9381 and 0.9258 RMSE values.

5.2 Structure Discovery. The hierarchical model can be used for classical pattern discovery tasks, such as the identification of the most appreciated items for each user community, as well a new kind of analysis, in which we focus on different topics and their impacts on the rating behaviour of users within the same community. Table 4 shows a selection from the most significant items for 10 user communities and their topics. We show 5 communities only, and the 5 most relevant topics within them. An item i is considered significant with respect to a topic h within the community k if $P_k(d_h|i) > P_k(d'_h|i) \forall h' \neq h$. For each community we register its prior probability (in square brackets) and the a-posteriori interpretation of its topics.

For instance, user community #2 is characterized by the topics: “Fantasy”, “Sci-Fi”, “Live-Music Performance” “Action” and “Drama”. It is worth noticing how the informative content in the hierarchy allows to better discriminate among topics and tendencies. By focusing on the first level only, the same community would exhibit a global attitude towards action movies (as “Gladiator”, “Die Hard” and “Terminator 2” are the most probable items here).

6 Conclusions and Future Works

We proposed a probabilistic model for the discovery of both global and local patterns from users’ preference data. Experimental evaluation showed that both the User Community model (for the discovery of global patterns) and the hierarchical topic detection model (for the discovery of local patterns) exhibits prediction capabilities comparable to state-of-the art approaches. Also, the proposed approach exhibits high flexibility in discovering structural patterns capable of providing suitable interpretations of the users’ preference data.

The proposed approach is suitable for further investigations in several respects. Foremost, the proposed strategy can be combined with temporal information in order to better model user changes in preferences. Also, the proposed approach allows suitable integration of prior modeling and Bayesian estimation, for the “cold-start” issues.

References

- [1] D. Goldberg *et al.*, “Using collaborative filtering to weave an information tapestry,” *Communications of ACM*, vol. 35, no. 12, pp. 61–70, 1992.
- [2] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, “Item-based collaborative filtering recommendation algorithms,” in *WWW ’01*, pp. 285–295.
- [3] S. Funk, “Netflix update: Try this at home,” 2006. [Online]. Available: <http://sifter.org/simon/journal/20061211.html>

	Community 1 [0.09]	Community 2 [0.05]	Community 3 [0.17]	Community 4 [0.06]	Community 5 [0.04]
Topic 1	Curb Your-Enthusiasm The Office: Series 2 The Office Special Monty Python's Flying Circus	The Incredibles The Princess Bride Lord of the Rings: The Two Towers	Gladiator The Shield Star Wars: EpiV Saving-Private Ryan	The Best-of Friends Friends: S6 Gilmore Girls Friends: S5	It's a Wonderful Life Star Wars: EpiV Ben-Hur Gone with the Wind
Interpr.:	Comedy	Fantasy	Action, war	Sitcom	Classic
Topic 2	Bruce Springsteen: Anthology 1978-2000 Karajan: Mozart: Don Giovanni Music of the Heart Music for Montserrat	Doctor Who: Pyramids of Mars Doctor Who: The Ribos Operation Battlestar Galactica Last Exile	Knowing Me-Knowing You Shag Aladdin Side Out	The Life and-Times of Frida Kahlo Birth of the-Blues / Blue Skies Julius Caesar American Dream SMM	Blue's Clues: Shapes and Colors Yu-Gi-Oh! Sesame Street Black Beauty
Interpr.:	Music	Sci-Fi	Comedy	Documentary	For children
Topic 3	Glengarry-Glen Ross JFK Bataan Changing Lanes	Harry Connick Jr.: Only You Donna Summer: Live Ben Harper: Live Mozart: Don Giovanni	The Secret-Lives of Dentists Proof of Life The Ice Storm Body Story	Reservoir Dogs Get Shorty The Naked Gun SMM	Gone in-60 Seconds Intolerable Cruelty Confidence The Naked Gun
Interpr.:	Drama	Live performances	Drama	Crime	Crime
Topic 4	Highlander The Recruit Ali Rambo: First Blood	Robin Hood: Prince of Thieves Proof of Life Mission Impossible II Vanilla Sky	Equilibrium Ladder 49 Bad Company Waking Life	Amelie Victor / Victoria Princess Mononoke Sophie's Choice	A Midsummer-Night's Dream Chances Are Fools Rush In Mighty Aphrodite
Interpr.:	Action	Action, famous actors	Thriller	Romance	Comedy, Fantasy
Topic 5	Men in Black Alien Resurrection Spider-Man 2 X-Men: Evolution	Love Story Coffee and Cigarettes A Walk in the Clouds Hannah and-Her Sisters	13 Going on 30 Planet of the Apes Men in Black Rosemary-and Thyme	All the Pretty Horses Romeo Must Die Great Expectations The Manchurian-Candidate	The Parallax View Waterworld Romeo Must Die Swimming-with Sharks
Interpr.:	Action, Sci-Fi	Drama, romance	Fantasy, Comedy	Drama	Thriller

Table 4: User communities and relevant topics

- [4] P. Arkadiusz, "Improving regularized singular value decomposition for collaborative filtering," in *SIGKDD'07*, pp. 39–42.
- [5] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *KDD '08*, pp. 426–434.
- [6] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *NIPS '08*, pp. 1257–1264.
- [7] D. Agarwal and B.-C. Chen, "flda: matrix factorization through latent dirichlet allocation," in *WSDM '10*, pp. 91–100.
- [8] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using markov chain monte carlo," in *ICML '08*, pp. 880–887.
- [9] D. H. Stern, R. Herbrich, and T. Graepel, "Matchbox: large scale online bayesian recommendations," in *WWW '09*, pp. 111–120.
- [10] I. Porteous, E. Bart, and M. Welling, "Multi-hdp: a non parametric bayesian model for tensor factorization," in *AAAI'08*, pp. 1487–1490.
- [11] P. Resnick *et al.*, "GroupLens: an open architecture for collaborative filtering of netnews," in *CSCW '94*, pp. 175–186.
- [12] T. George and S. Merugu, "A scalable collaborative filtering framework based on co-clustering," in *ICDM '05*, pp. 625–628.
- [13] T. Hofmann and J. Puzicha, "Latent class models for collaborative filtering," in *IJCAI'99*, pp. 688–693.
- [14] R. Jin, L. Si, and C. Zhai, "A study of mixture models for collaborative filtering," *Information Retrieval*, vol. 9, no. 3, pp. 357–382, 2006.
- [15] M. Khoshneshin and W. N. Street, "Incremental collaborative filtering via evolutionary co-clustering," in *RecSys '10*, pp. 325–328.
- [16] N. Barbieri, M. Guarascio, and G. Manco, "A block mixture model for pattern discovery in preference data," in *TFDOM '10 (ICDM workshop)*.
- [17] R. M. Bell and Y. Koren, "Scalable collaborative filtering with jointly derived neighborhood interpolation weights," in *ICDM '07*, pp. 43–52.
- [18] R. Bell, Y. Koren, and C. Volinsky, "Modeling relationships at multiple scales to improve accuracy of large recommender systems," in *KDD '07*, pp. 95–104.
- [19] B. Marlin, "Modeling user rating profiles for collaborative filtering," in *NIPS'03*.
- [20] —, "Collaborative filtering: A machine learning perspective," Master's thesis, Department of Computer Science, University of Toronto, 2004.
- [21] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *The Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.
- [22] D. M. Blei and J. D. McAuliffe, "Supervised topic models," in *NIPS '08*.
- [23] D. Agarwal and B.-C. Chen, "Regression-based latent factor models," in *KDD '09*, pp. 19–28.
- [24] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *KDD '08*, pp. 650–658.
- [25] N. D. Lawrence and R. Urtasun, "Non-linear matrix factorization with gaussian processes," in *ICML '09*, pp. 601–608.
- [26] A. Banerjee *et al.*, "A generalized maximum entropy approach to bregman co-clustering and matrix approximation," in *KDD '04*, pp. 509–514.
- [27] T. Hofmann, "Collaborative filtering via gaussian probabilistic latent semantic analysis," in *SIGIR '03*, pp. 259–266.
- [28] Y. Koren, "Collaborative filtering with temporal dynamics," in *KDD '09*, pp. 447–456.