An Analysis of Probabilistic Methods for Top-N Recommendation in Collaborative Filtering

No Institute Given

Abstract. In this work we perform an analysis of probabilistic approaches to recommendation upon a different validation perspective, which focuses on accuracy metrics such as recall and precision of the recommendation list. Traditionally, state-of-art approches to recommendations consider the recommendation process from a "missing value prediction" perspective. This approach simplifies the model validation phase that is based on the minimization of standard error metrics such as RMSE. However, recent studies have pointed several limitations of this approach, showing that a lower RMSE does not necessarily imply improvements in terms of specific recommendations. We demonstrate that the underlying probabilistic framework offers several advantages over traditional methods, in terms of flexibility in the generation of the recommendation list and consequently in the accuracy of recommendation.

Keywords: Recommender Systems, Collaborative Filtering, Probabilistic Topic Models, Performance

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. Recommendations can rely either on static information about the content of the available catalogs [16], or on a-posteriori analysis of past behavior through collaborative filtering approaches (CF) [7]. CF techniques are effective with huge catalogs when information about past interactions is available.

To improve the accuracy of CF-based recommendation engines, researchers have focused on the development of accurate techniques for rating prediction. The recommendation problem has been interpreted as a missing value prediction problem [19], in which, given an active user, the system is asked to predict her preference for a set of items. Since a user is more prone to access items for which she will likely provide a positive feedback, a recommendation list can be be hence built by drawing upon the (predicted) highly-rated items.

Under this perspective, a common approach to evaluate the predictive skills of a recommender systems is to minimize statistical error metrics, such as the *Root Mean Squared Error (RMSE)*. The common assumption is that small improvements in RMSE would reflect into an increase of the accuracy of the recommendation lists. This assumption, however does not necessarily hold. In [4], the authors review the most common approaches to CF-based recommendation, and compare them according to a new testing methodology which focuses on the accuracy of the recommendation lists rather than on the rating prediction accuracy. Notably, cutting-edge approaches characterized by low RMSE values achieves performances comparable to naive techniques, whereas simpler approaches, such as the pure SVD, consistently outperforms the other techniques. In an attempt to find an explanation, the authors impute the contrasting behavior with a "limitation of RMSE testing, which concentrates only on the ratings that the user provided to the system" and consequently "misses much of the reality, where all items should count, not only those actually rated by the user in the past" [4].

The point is that pure SVD rebuilds the original rating matrix in terms of latent factors, rather than trying to minimize the error on observed data. In practice, the underlying optimization problem is quite different, since it takes into account the whole rating matrix considering both observed and unobserved preference values. To summarize, it is likely to better identify the latent factors and the hidden relationships between both factor/users and factors/items. It is natural then to ask whether more sophisticated latent factor models confirm this trend, and are able to guarantee better results in terms of recommendation accuracy, even when they provide poor RMSE performances.

Among the state-of-the art latent factor models, probabilistic techniques offer some advantages over traditional deterministic models: notably, they do not minimize a particular error metric but are designed to maximize the likelihood of the model given the data which is a more general approach; moreover, they can be used to model a distribution over rating values which can be used to determine the confidence of the model in providing a recommendation; finally, they allow the possibility to include prior knowledge into the generative process, thus allowing a more effective modeling of the underlying data distribution. However, previous studies on recommendation accuracy do not take into consideration such probabilistic approaches to CF, which instead appear rather promising under the above devised perspective.

In this paper we adopt the testing methodology proposed in [4], and discuss also other metrics [6] for assessing the accuracy of the recommendation list. Based on these settings, we perform an empirical study of some paradigmatic probabilistic approaches to recommendation. We study different techniques to rank items in a probabilistic framework, and evaluate their impact in the generation of a recommendation list. We shall consider approaches for both implicit and explicit preference values, and show that latent factor models, equipped with the proper ranking functions, achieve competitive advantages over traditional techniques.

The rest of the paper is organized as follows: the testing methodology and the accuracy metrics are discussed in Sec. 2. Section 3 introduces the probabilistic approaches to CF that we are interested in evaluating. The approaches we include can be considered representative of wider classes which share the same roots. In this context, our results can be extended to more sophisticated approaches. Finally, in Sec. 4 we compare the approaches and assess their effectiveness according to the selected testing methodology.

2 Evaluating Recommendations: A Review

To begin with, we introduce some notation to be used throughout the paper. User's preferences can be represented by using a $m \times n$ rating matrix $\mathbf{R}$, where m is the cardinality of the *user-set* $\mathcal{U}_{\mathbf{R}} = \{u_1, \cdots, u_n\}$ and n is the cardinality of the *item-set* $\mathcal{I}_{\mathbf{R}} = \{i_1, \cdots, i_n\}$. We denote by r_i^u (resp. $\mathbf{R}_i^u$ when the reference to the matrix needs to be made explicit) the rating value associated to the pair $\langle u, i \rangle$. Values fall within a fixed integer range $\mathcal{V} = \{0, \cdots, V\}$, where 0 denotes "rating unknown", and V represents the highest interest value . Implicit feedback assumes that $V = 1$. When $V > 1$, we will denote by $\overline{r}_{\mathbf{R}}$ the average rating among all those ratings $r_i^u > 0$ in $\mathbf{R}$. Users tend to express their interest only on a restricted number of items; thus, the rating matrix is characterized by an exceptional sparseness factor (e.g., more than 95%).. Let $\mathcal{I}_{\mathbf{R}}(u)$ denotes the set of products rated by the user u: $\mathcal{I}_{\mathbf{R}}(u) = \{i \in \mathcal{I} : r_i^u \neq 0\}$; symmetrically, we will denote by $\mathcal{U}_{\mathbf{R}}(i)$ the set of users who have expressed their preference on the item i.

The general framework for the generation of a recommendation list can be modeled as follows. We will denote by $\mathcal{L}_u^j$ the recommendation list provided by the system to the user u during a generic session j. Then, the following protocols applies:

- Let $\mathcal{C}_u^j$ a list of D candidate random items unrated by the user u in the past sessions $1, \ldots, j-1$;
- Associate to each item $i \in \mathcal{C}_u^j$ a score $p_i^{u,j}$ which represents the user's interest for i in session j;
- Sort $\mathcal{C}_u^j$ in descending order given the values $p_i^{u,j}$;
- Add the first N items from $\mathcal{C}_u^j$ to $\mathcal{L}_u^j$ and return the latter to the user.

Simple scoring functions can be obtained considering non-personalized baseline models which take into account the popularity or the average rating of an items. More specifically, *Top Popular (Top-Pop)* recommends items with the highest number of ratings, while *Item Average (Item-Avg)* selects items with the highest average rating. For the purposes of this paper, we assume that each RS is capable of providing a specific scoring $p_i^{u,j}$. Thus, the testing methodology basically relies on the evaluation of capability of the RS in providing higher scores for the items of interest in $\mathcal{C}_u^j$.

A common framework in the evaluation of the predictive capabilities of a RS algorithm is to split the rating matrix $\mathbf{R}$ into matrices $\mathbf{T}$ and $\mathbf{S}$: the first one is used to train the RS, while the latter is used for validation. It is worth noticing that, while both $\mathbf{T}$ and $\mathbf{S}$ share the same dimensions as $\mathbf{R}$, for each pair (u, i) we have that $\mathbf{S}_i^u > 0$ implies $\mathbf{T}_i^u = 0$, i.e. no incompatible values overlap between training and test set. By selecting a user in $\mathbf{S}$, the set $\mathcal{C}_u^j$ is obtained by drawing upon $\mathcal{I}_{\mathbf{R}} - \mathcal{I}_{\mathbf{T}}(u)$. Next, we ask the system to predict a set of items which he/she may like and then measure the accuracy of the provided recommendation. Here, the accuracy is measured by comparing the top-N items selected by resorting to the RS, with those appearing in $\mathcal{I}_{\mathbf{S}}(u)$.

Precision and Recall of the Recommendation List. A first, coarse-grained approach to evaluation, can be obtained by employing standard classification-based accuracy metrics such as precision and recall, which require the capability to distinguish between relevant and not relevant recommendations. Given a user, we assume a unique session of recommendation, and we compare the recommendation list of N items provided by the RS, according to the protocol described above, with those relevant items in $\mathcal{I}_{\mathbf{S}}(u)$. In particular, assuming we can identify a subset $\mathcal{T}_u^r \subseteq \mathcal{I}_{\mathbf{S}}(u)$ of relevant items, we can compute precision and recall as:

$$Recall(N) = \frac{1}{M} \sum_{u=1}^{M} \frac{|\mathcal{L}_u^{\cdot} \cap \mathcal{T}_u^r|}{|\mathcal{T}_u^r|}$$

$$Precision(N) = \frac{1}{M} \sum_{u=1}^{M} \frac{|\mathcal{L}_u^{\cdot} \cap \mathcal{T}_u^r|}{N}$$

Relevance can be measured in several different ways. Here we adopt two alternative definitions. When $V > 1$ (i.e., an explicit preference value is available) we denote as relevant all those items which received a rating greater than the average ratings in the training set, i.e.,

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

Implicit preferences assume instead that all items in $\mathcal{I}_{\mathbf{S}}(u)$ are relevant.

Evaluating Users Satisfaction. The above definitions of precision and recall aims at evaluating the amount of useful recommendations in a single session. A different perspective can be considered by assuming that a recommendation meets user satisfaction if he/she can find in the recommendation list at least an item which meets his/her interests. This perspective can be better modeled by a different approach to measure accuracy, as proposed in [5, 4]. The approach relies on a different definition of relevant items, namely:

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

Then, the following testing protocol can be applied:

- For each user u and for each positively-rated item $i \in \mathcal{T}_u^r$:
  - Generate the candidate list $\mathcal{C}_u$ by randomly drawing from $\mathcal{I}_{\mathbf{R}} - (\mathcal{I}_{\mathbf{T}}(u) \cup \{i\})$;
  - add i to $\mathcal{C}_u$ and sort the list according to the scoring function;
  - Record the position of the item i in the ordered list:
    - if i belongs to the top-k items, we have a *hit*
    - otherwise, we have a *miss*

Practically speaking, we ask the RS to rank an initial random sample which also contains i. If i is actually recommended, we have an hit, otherwise the RS has failed in detecting an item of high interest for the considered user. Recall and precision can hence be tuned accordingly:

$$US_{Recall(N)} = \frac{\#hits}{|T_s|} \tag{1}$$

$$US_{Precision(N)} = \frac{\#hits}{N \cdot |\mathcal{T}_u^r|} = \frac{recall(N)}{N} \tag{2}$$

Notice that the above definition of precision does not penalize false positives: the recommendation is considered successful if it matches at least an item of interest. However, neither the amount of non-relevant“spurious” items, nor the position of the relevant item within the top-N is taken into account.

3 Collaborative Filtering in a Probabilistic Framework

Probabilistic approaches assume that each preference observation is randomly drawn from the joint distribution of the random variables which model users, items and preference values (if available). Typically, the random generation process follows a bag of words assumption and preference observations are assumed to be generated independently. A key difference between probabilistic and deterministic models relies in the inference phase: while the latter approaches try to minimize directly the error made by the model, probabilistic approaches do not focus on a particular error metric; parameters are determined by maximizing the likelihood of the data, typically employing an Expectation Maximization procedure. In addition, background knowledge can be explicitly modeled by means prior probabilities, thus allowing a direct control on overfitting within the inference procedure [8]. By modeling prior knowledge, they implicitly solve the need for regularization which affects traditional gradient-descent based latent factors approaches.

Further advantages of probabilistic models can be found in their easy interpretability: they can often be represented by using a graphical model, which summarizes the intuition behind the model by underlying causal dependencies between users, items and hidden factors. Also, they provide an unified framework for combining collaborative and content features [1, 21, 11], to produce more accurate recommendations even in the case of new users/items. Moreover, assuming that an explicit preference value is available, probabilistic models can be used to model a distribution over rating values which can be used to infer confidence intervals and to determine the confidence of the model in providing a recommendation.

In the following we will briefly introduce some paradigmatic probabilistic approaches to recommendation, and discuss how these probabilistic model can be used for item ranking, which is then employed to produce the top-N recommendation list. The underlying idea of probabilistic models based on latent factors is that each preference observation $\langle u, i\rangle$ is generated by one of k possible states, which informally model the underlying reason why u has chosen/rated i. Based on the mathematical model, two different inferences can be then supported to be exploited in item ranking, where the main difference [9], lies in a difference way of modeling data according to the underlying model:

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

3.1 Modeling Preference Data

In the simplest model, we assume that a user u is associated with a latent factor Z, and ratings for an item i are generated according to this factor. the generative model for this mixture is given in Fig. 1(a). The θ parameter here a the prior probability distribution $P(Z)$, whereas $\beta_{i,z}$ is the prior for the rating generation $P(R=r|i,z)$. We shall refer to the **Multinomial Mixture Model** (MMM, [14]) to denote that $\beta_{i,z}$ is a multinomial over $\mathcal{V}$. Forced prediction can be achieved by

$$P(r|i,u) = \sum_z \beta_{z,i,r} P(z|u) \tag{3}$$

where

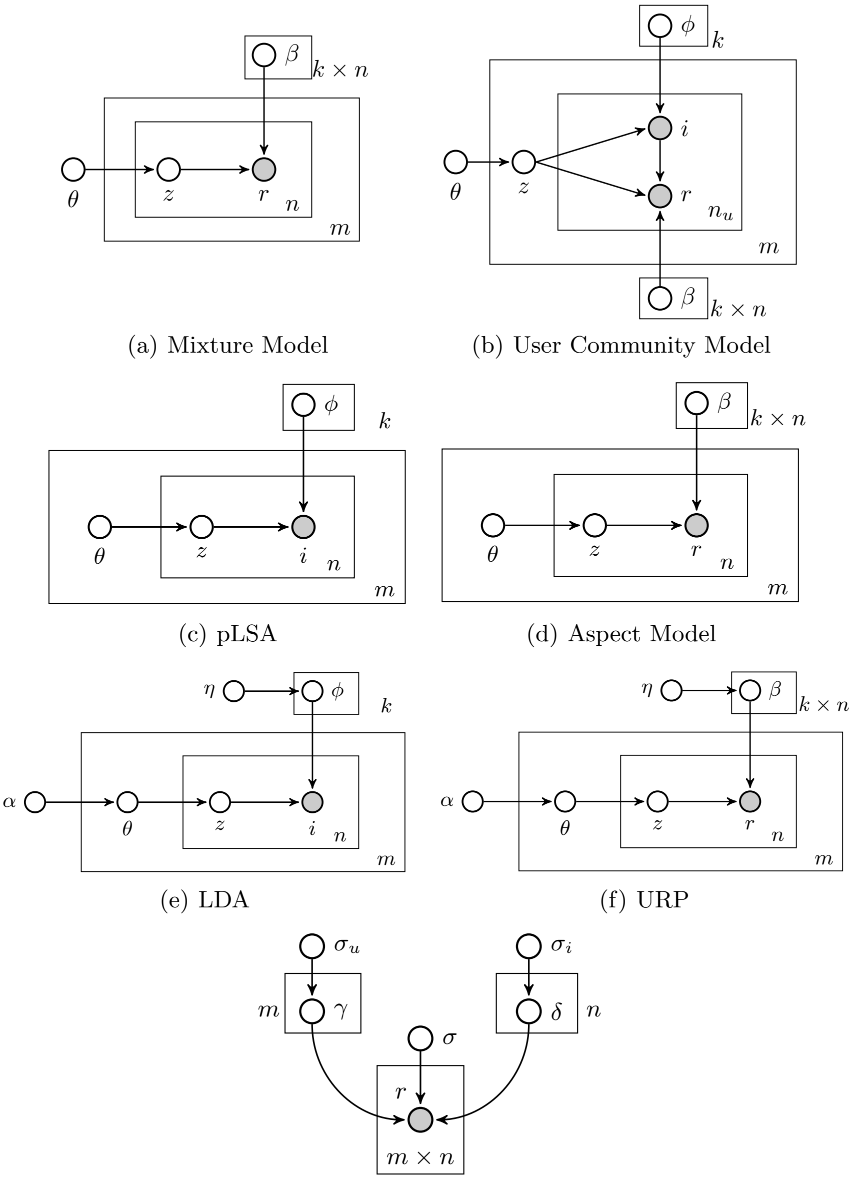
$$P(z|u) \propto P(\mathbf{u}_{obs}|z)\theta_z$$

and $\mathbf{u}_{obs}$ represents the observed values (u,i,r) in $\mathbf{R}$.

The **probabilistic Latent Semantic Analysis** approach (PLSA, [9]) specifies a co-occurence data model in which the user u and item i are conditionally independent given the state Z of the latent factor. Differently from the previous mixture model, where a single latent factor is associated with every user u, the PLSA model associates a latent variable with every observation triplet (u,i,r). Hence, different ratings of the same user can be explained by different latent causes in PLSA (modeled as priors $\{\theta_u\}_{1,\ldots,m}$ in Fig. 1(c)), whereas a mixture model assumes that all ratings involving the same user are linked to the same underlying community. PLSA directly supports item selection:

$$P(i|u) = \sum_z \phi_{z,i}\theta_{u,z} \tag{4}$$

where ϕ_z represents a multinomial distribution over items. The main drawback of the PLSA approach is that it cannot directly model new users, because the parameters $\theta_{u,z} = P(z|u)$ are specified only for those users in the training set.



(a) Mixture Model

(b) User Community Model

(c) pLSA

(d) Aspect Model

(e) LDA

(f) URP

(g) Probabilistic Matrix Factorization

Fig. 1. Generative models to preference data.

We consider two further variants for the PLSA, where explicit preferences are modeled by an underlying distribution $\beta_{z,i}$. In the **Aspect Model** (AM, [10]) $\beta_{z,i}$ is a multinomial over $\mathcal{V}$. In this case, the rating probability can be modeled as

$$P(r|u,i) = \sum_z \beta_{r,i,z}\theta_{u,z} \tag{5}$$

Conversely, the **Gaussian Mixture Model** (G-PLSA, [8]) models $\beta_{z,i} = (\mu_{iz}, \sigma_{iz})$ as a gaussian distribution, and provides a normalization of ratings through the user's mean and variance, thus allowing to model users with different rating patterns. The corresponding rating probability is

$$P(r|u,i) = \sum_z \mathcal{N}(r; \mu_{iz}, \sigma_{iz})\theta_{u,z} \tag{6}$$

The **Latent Dirichlet Allocation** [3] is designed to overcome the main drawback in the PLSA-based models, by introducing Dirichlet priors, which provide a full generative semantic at user level and avoid overfitting. Again, two different formulations, are available, based on whether we are interested in modeling implicit (LDA) or explicit (**User Rating Profile**, URP[13]) preference values. In the first case, we have:

$$P(i|u) = \int \sum_z \phi_{z,i}\theta_z P(\theta|\mathbf{u}_{obs})d\theta \tag{7}$$

(where $P(\theta|\mathbf{u}_{obs})$ is estimated in the inference phase). Analogously, for the URP we have

$$P(r|u,i) = \int \sum_z \beta_{z,i,r}\theta_z P(\theta|\mathbf{u}_{obs})d\theta \tag{8}$$

The **User Communities Model** (UCM, [2]) adopts the same inference formula Eq. 3 of the multinomial model. Nevertheless, it introduces some key features, that combine the advantages of both the AM and the MMM, as shown in Fig. 1(b). First, the exploitation of a unique prior distribution θ over the user communities helps in preventing overfitting. Second, adds flexibility in the prediction by modeling an item as an observed (and hence randomly generated) component. UCM directly a free-prediction approach.

Finally, the **Probabilistic Matrix Factorization** approach (PMF, [18]) reformulates the rating assignment as a matrix factorization. Given the latent user and item k-feature matrices γ_u and δ_i, (where K denotes the number of the features employed in the factorization), the preference value is generated by assuming a Gaussian distribution over rating values conditioned on the interactions between the user and the considered item in the latent space, as shown in Fig. 1(g). In practice, $P(r|u,i)$ is modeled as a gaussian distribution, with mean $\gamma_u^T\delta_i$ and fixed variance σ:

$$P(r|u,i) = \mathcal{N}(r; \gamma_u^T\delta_i, \sigma^2) \tag{9}$$

Both the original approach and its bayesian generalizations [17, 20] are characterized by high prediction accuracy.

3.2 Item Ranking

In this section we discuss how the above described models can be used to provide the ranking p_i^u for a given user u and an item i in the protocol described in Sec. 2.

Predicted Rating. The most intuitive way to provide item ranking in the recommendation process relies on the analysis of the distribution over preference values $P(r|u,i)$ (assuming that we are modeling explicit preference data). Given this distribution, there are several methods for computing the ranking for each pair $\langle u,i \rangle$; the most commonly used is the expected value $E[R|u,i]$, as it minimizes the MSE and thus the RMSE:

$$p_i^u = E[R|u,i] \tag{10}$$

We will show in Sec. 4 that this approach fails in providing accurate recommendation and discuss about potential causes.

Item Selection. For co-occurrence preference approaches, the rank of each item i, with regards to the user u can be computed as the mixture:

$$p_i^u = P(i|u) = \sum_z P(z|u)P(i|z) \tag{11}$$

where $P(i|z)$ is the probability that i will be selected by users represented by the abstract pattern z. This distribution is a key feature of co-occurrence preference approaches and models based on free-prediction. When $P(i|z)$ is not directly inferred by the model, we can still estimate it by averaging on all the possible users who selected i:

$$P(i|z) \propto \sum_u \delta(u,i)_{\mathbf{T}} P(z|u)$$

where $\delta_{\mathbf{T}}(u,i) = 1$ if $\mathbf{T}_i^u \neq 0$.

Item Selection And Relevance. In order to force the selection process to concentrate on relevant items, we can extend the ranking discussed above, by including a component that represents the "predicted" relevance of an item with respect to a given user:

$$\begin{aligned} p_i^u &= P(i, r > \overline{r}_{\mathbf{T}}|u) \\ &= P(i|u)P(r > \overline{r}_{\mathbf{T}}|u,i) = \sum_z P(z|u)P(i|z)P(r > \overline{r}_{\mathbf{T}}|i,z) \end{aligned} \tag{12}$$

where $P(r > \overline{r}_{\mathbf{T}}|i,z) = \sum_{r > \overline{r}_{\mathbf{T}}} P(r|i,z)$. In practice, an item is ranked on the basis of the value of its score, by giving high priority to the high-score items.

4 Evaluation

In this section we experiment the testing protocols presented in Sec. 2 on the probabilistic approaches defined in the previous section. We use the MovieLens-1M[1] dataset, which consists of $1,000,209$ ratings given by $6,040$ users on approximately $3,706$ movies, with a sparseness coefficient 96% and an average number of ratings 132 per user, and 216 per item. In the evaluation phase, we adopt a MonteCarlo 5-folds validation, where for each fold contains about the 80% of overall ratings and the remaining data (20%) is used as test-set. The final results reported by averaging the values achieved in each fold.

In order to make our results comparable with the ones reported in [4], we consider *Top-Pop* and *Item-Avg* algorithms as baseline, and *Pure-SVD* as a main competitor. Notice that there are some differences between our evaluation and the one performed in the above cited study, namely: (i) we decided to employ bigger test-sets (20% of the overall data vs 1.4%) and to cross-validate the results; (ii) for lack of space we concentrate on MovieLens only, and omit further evaluations on the Netflix data (which however, in the original paper [4], confirm Pure-SVD as the top-performer); (iii) we decided to omit the “long tail” test, aimed at evaluating the capability of suggesting non-trivial items, as it is out of the scope of this paper.[2]

In the following we study the effects of the ranking function on the accuracy of the recommendation list. The results we report are obtained by varying the length of the recommendation list in the range $1 - 20$ and the dimension of the random sample is fixed to $D = 1000$. In a preliminary test, we found the optimal number of components for the *Pure-SVD* to be set to 50.

4.1 Expected Value

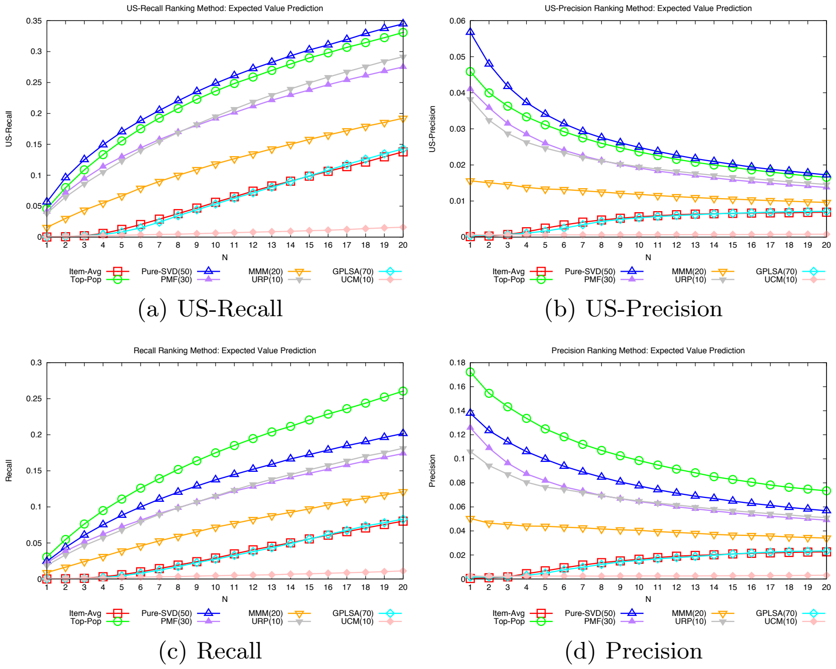
We start our analysis from the evaluation of the recommendation accuracy achieved by approaches that model explicit preference data, namely PMF, MMM, URP, UCM and G-PLSA, where the predicted rating is employed as ranking function. First of all, the following table summarizes the RMSE obtained by these approaches:

Approach	**RMSE**	**#Latent Factors**
Item Avg	0.9784	-
MMM	1.0000	20
G-PLSA	0.9238	70
UCM	*0.9824*	10
URP	*0.8989*	10
PMF	0.8719	30

[1] http://www.grouplens.org/system/files/ml-data-10M100K.tar.gz

[2] Notice, however, that it is still possible to perform an indirect measurement of the non-triviality and correctness of the discussed approaches by measuring the gain in recommendation accuracy wrt. the Top-Pop recommendation algorithm.

The results about Recall and Precision are given in Fig. 2, where the respective number of latent factors is given in brackets. Considering user satisfaction, almost all the probabilistic approaches fall between the two baselines. Pure-SVD outperforms significantly the best probabilistic performers, namely URP and PMF. The trend for probabilistic approaches does not change considering Recall and Precision, but in this case not even the Pure-SVD is able to outperform Top-Pop, which exhibits a consistent gain over all the considered competitors.



(a) US-Recall

(b) US-Precision

(c) Recall

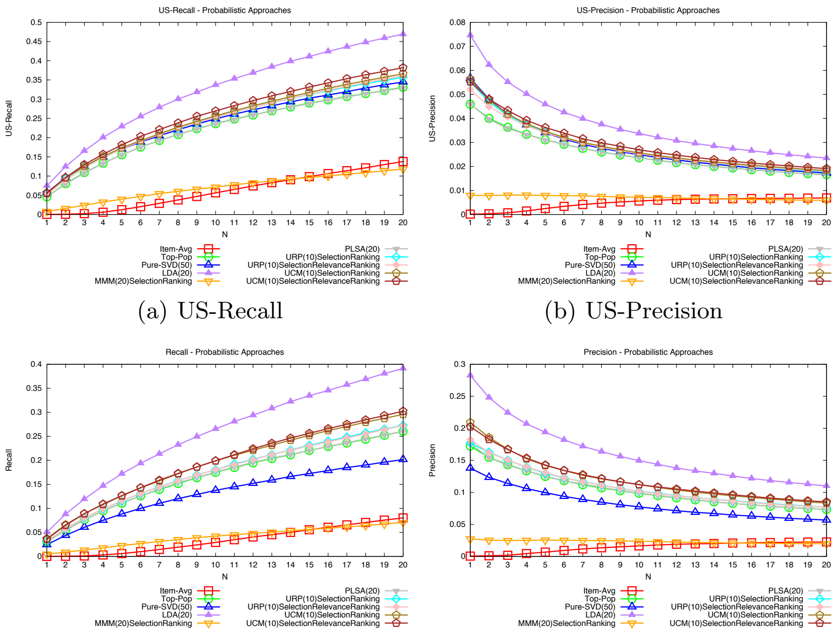
(d) Precision

Fig. 2. Recommendation Accuracy achieved by probabilistic approaches considering $E[r|u,i]$ as ranking function

A first summary can be obtained as follows. First, we can confirm that there is no monotonic relationship between RMSE and recommendation accuracy. All the approaches tend to have a non-deterministic behavior, and even the best approaches provide unstable results depending on the size N. Further, ranking by the expected value exhibits unacceptable performance on the probabilistic approaches, which reveal totally inadequate in this perspective. More in general, any variant of this approach that we do not report here for space limitations) does not substantially change the results.

4.2 Item Occurrence

Things radically change when item occurrence is taken into consideration. Fig. 3 show the recommendation accuracy achieved by probabilistic models which employ Item-Selection (LDA,PLSA,UCM and URP) and Item-Selection&Relevance (UCM and URP). The LDA approach significantly outperforms all the available approaches. Surprisingly, UCM is the runner-up, as opposed to the behavior exhibited with the expected value ranking. it is clear that the component $P(i|z)$ here plays a crucial role, that is further strengthened by the relevance ranking component.



(a) US-Recall

(b) US-Precision

(c) Recall

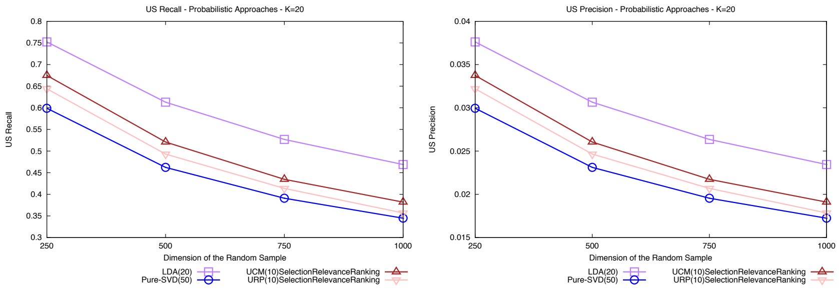
(d) Precision

Fig. 3. Recommendation Accuracy achieved by probabilistic approaches considering $P(i|u)$ or $P(i, r > 3|u)$ as ranking functions

Also surprising is the behavior of URP, which still achieves a satisfactory performance compared to Pure-SVD. However, it does not compare to LDA. The reason can be found in the fact that the inference procedure in the LDA directly estimates $P(i|z)$, whereas such a component in the URP model is approximated a-posteriori. This is also proved by the unsatisfactory performance of the MMM approach which falls short of the expectations. Since the UCM is an extension of the MMM, it is clear that explicitly inferring the ϕ component in the model helps in achieving a stronger accuracy.

The PLSA model also seems to suffer from from the overfitting issues, as it is not able to reach the performances of the Pure-SVD. On the other side, if user satisfaction is not taken into account, the PLSA outperforms the Pure-SVD, as it follows the general trend of the Top-Pop model. More in general, models equipped with Item-Selection&Relevance outperform their respective version which make recommendation basing only on the Item-Selection component.

We also perform an additional test to evaluate the impact of the size of the random sample in the testing methodology employed to measure user satisfaction. Results achieved by LDA,Pure-SVD, UCM/URP (Selection&Relevance Ranking) are given in Fig. 4. Probabilistic approaches outperform systematically Pure-SVD for each value of D.



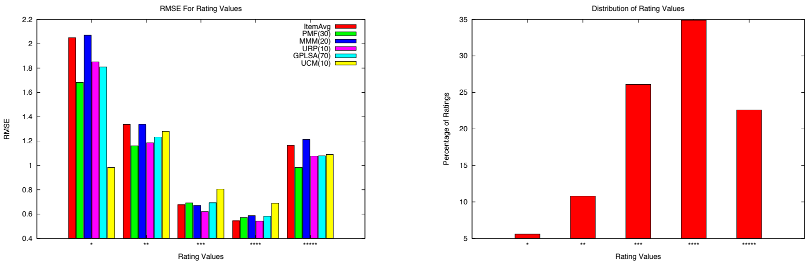
(a) US-Recall (b) US-Precision

Fig. 4. Recommendation Accuracy achieved by probabilistic approaches considering K=20 and varying the dimension of the random Sample

4.3 Discussion

There are two main considerations in the above figures. One is that rating prediction fails in providing accurate recommendations. The second observation is the unexpected strong impact of the item selection component, when properly estimated.

In an attempt to carefully analyze the rating prediction pitfalls, we can plot in Fig. 5(a) the contribution to the RMSE in each single evaluation in $\mathcal{V}$ by the probabilistic techniques under consideration. Item-Avg acts as baseline here. While predictions are accurate for values $3-4$, they result rather inadeguate for border values, namely $1, 2$ and 5. This is mainly due to the nature of RMSE, which penalizes larger errors. This clearly supports the thesis that low RMSE does not necessarily induces good accuracy, as the latter is mainly influenced by the items in class 5 (where the approaches are more prone to fail). It is clear that a better tuning of the ranking function should take this component into account.



(a) RMSE by rating value

(b) Rating values in MovieLens

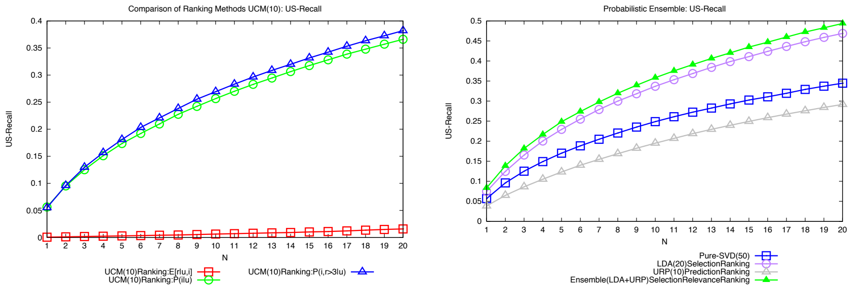
Fig. 5. Analysis of Prediction Accuracy

Also, by looking at the distribution of the rating values, we can see that the dataset is biased towards the mean values, and more in general the low rating values represent a lower percentage. This explains, on one side, the tendency of the expected value to flatten towards a mean value (and hence to fail in providing an accurate prediction). On the other side, the lack of low-rating values provides an interpretation of the dataset as a *Like/DisLike* matrix, for which the item selection tuning provides a better modeling.

By the way, the rating information, combined with item selection, provides a marginal improvement, as testified by Fig. 6(a). Here, a closer look at the UCM approach is taken, by plotting three curves relative to the three different approaches to item ranking. Large recommendation lists tend to be affected by the rating prediction.
Our experiments have shown that item selection component plays the most important role in recommendation ranking. However, better results can be achieved by considering also a rating prediction component. To empirically prove the effectiveness of such approach, we performed a final test in which item ranking is performed by employing an ensemble approach based on the item selection and relevance ranking. In this case, the components of the ranking come from different model: the selection probability is computed according to an LDA model, while the relevance ranking is computed by employing the URP model. Fig. 6(b) shows that this approach outperforms LDA, achieving the best result in recommendation accuracy (due to the lack of space we show only the trend correspoding to US-Recall).

5 Conclusion and Future Works

We have shown that probabilistic models, equipped with the proper ranking function, exhibit competitive advantages over state-of-the-art RS in terms of recommendation accuracy. In particular, we showed strategies based on item selection guarantee significant improvements, and we have investigated the motivations



(a) Comparison of Different Ranking Function on UCM(10)

(b) Probabilistic Ensamble

behind the failure of prediction-based approaches. The advantage of probabilistic models lies in their flexibility, as they allow switching between both methods in the same inference framework. The nonmonotonic behavior of RMSE also finds its explanation in the distribution of errors along the rating values, thus suggesting different strategies to be developed for providing prediction-based recommendation.

Besides the above mentioned, there are other significant advantages in the adoption of probabilistic models for recommendation. Recent studies pointed out that there is more in recommendation than just rating prediction. A successful recommendation should answer to the simple question 'What is the user actually looking for?' which is strictly tied with dynamic user profiling. Moreover, prediction-based recommender systems do not consider one of the most important applications from the retailer point of view: suggesting users products they would not have found otherwise discovered.

In [15] the authors argued that the popular testing methodology based on prediction accuracy is rather inadeguate and does not capture important aspects of the recommendations, like non triviality, serendipity, users needs and expectations, and their studies have scaled down the usefulness of achieving a lower RMSE [12]. In short, the evaluation of a recommender cannot rely exclusively on prediction accuracy but must take into account what is really displayed to user, i.e the recommendation list, and its impact on his/her navigation.

Clearly, probabilistic graphical models, like the ones discussed in this paper, provide several components which can be fruitfully exploited for the estimation of such measures. Latent factors, probability of item selection and rating probability can help in better specify usefulness in recommendation. We plan to extend the framework in this paper in this promising directions, by providing subjective measures for such features and measuring the impact of such models.

References

1. Agarwal, D., Chen, B.C.: flda: matrix factorization through latent dirichlet allocation. In: WSDM. pp. 91–100 (2010)
2. Barbieri, N., Guarascio, M., Manco, G.: A probabilistic hierarchical approach for pattern discovery in collaborative filtering data. In: SMD (2011)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. The Journal of Machine Learning Research 3, 993–1022 (2003)
4. Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-n recommendation tasks. In: ACM RecSys. pp. 39–46 (2010)
5. Cremonesi, P., Turrin, R., Lentini, E., Matteucci, M.: An evaluation methodology for collaborative recommender systems. In: AXMEDIS. pp. 224–231 (2008)
6. Ge, M., Delgado-Battenfeld, C., Jannach, D.: Beyond accuracy: evaluating recommender systems by coverage and serendipity. In: ACM RecSys. pp. 257–260 (2010)
7. Goldberg, D., Nichols, D., Oki, B.M., Terry, D.: Using collaborative filtering to weave an information tapestry. Communications of the ACM 35(12), 61–70 (1992)
8. Hofmann, T.: Collaborative filtering via gaussian probabilistic latent semantic analysis. In: SIGIR (2003)
9. Hofmann, T.: Latent semantic models for collaborative filtering. ACM Transactions on Information Systems (TOIS) 22(1), 89–115 (2004)
10. Hofmann, T., Puzicha, J.: Latent class models for collaborative filtering. In: IJCAI. pp. 688–693 (1999)
11. Jin, X., Zhou, Y., Mobasher, B.: A maximum entropy web recommendation system: combining collaborative and content features. In: KDD. pp. 612–617 (2005)
12. Koren, Y.: How useful is a lower rmse? (2007), http://www.netflixprize.com/community/viewtopic.php?id=828
13. Marlin, B.: Modeling user rating profiles for collaborative filtering. In: NIPS (2003)
14. Marlin, B., Marlin, B.: Collaborative filtering: A machine learning perspective. Tech. rep., Department of Computer Science University of Toronto (2004)
15. McNee, S., Riedl, J., Konstan, J.A.: Being accurate is not enough: How accuracy metrics have hurt recommender systems. In: ACM SIGCHI Conference on Human Factors in Computing Systems. pp. 1097–1101 (2006)
16. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. The adaptive web: methods and strategies of web personalization pp. 325–341 (2007)
17. Salakhutdinov, R., Mnih, A.: Bayesian probabilistic matrix factorization using markov chain monte carlo. In: ICML. pp. 880–887 (2008)
18. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: NIPS. pp. 1257–1264 (2008)
19. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-based collaborative filtering recommendation algorithms. In: WWW. pp. 285–295 (2001)
20. Shan, H., Banerjee, A.: Generalized probabilistic matrix factorizations for collaborative filtering. In: ICDM (2010)
21. Stern, D.H., Herbrich, R., Graepel, T.: Matchbox: large scale online bayesian recommendations. In: WWW. pp. 111–120 (2009)