



Universidade de São Paulo

Biblioteca Digital da Produção Intelectual - BDPI

Departamento de Ciências de Computação - ICMC/SCC

Comunicações em Eventos - ICMC/SCC

2014-11

Ensemble learning in recommender systems: combining multiple user interactions for ranking personalization

Brazilian Symposium on Multimedia and the Web, 20th, 2014, João Pessoa.

<http://www.producao.usp.br/handle/BDPI/48609>

Downloaded from: Biblioteca Digital da Produção Intelectual - BDPI, Universidade de São Paulo

Ensemble Learning in Recommender Systems: Combining Multiple User Interactions for Ranking Personalization

Arthur Fortes and Marcelo Manzano
Institute of Mathematics and Computer Science
University of São Paulo
São Carlos, SP, Brazil
{fortes,mmanzato}@icmc.usp.br

ABSTRACT

In this paper, we propose a technique that uses multimodal interactions of users to generate a more accurate list of recommendations optimized for the user. Our approach is a response to the actual scenario on the Web which allows users to interact with the content in different ways, and thus, more information about his preferences can be obtained to improve recommendation. The proposal consists of an ensemble learning technique that combines rankings generated by unimodal recommenders based on particular interaction types. By using a combination of different types of feedback from users, we are able to provide better recommendations, as shown by our experimental evaluation.

Categories and Subject Descriptors

H.3.1 [Information Systems]: Information storage and retrieval—*Content analysis and Indexing*; I.2.6 [Computing Methodologies]: Artificial Intelligence—*Learning*

General Terms

Algorithms; Ensemble Learning

Keywords

Recommender Systems; Ensemble Learning; Multimodal Interactions

1. INTRODUCTION

As the exponential growth of information generated on the World Wide Web, Information Filtering techniques like Recommender Systems have become more and more important and popular. Recommender systems consist of a specific type of information filtering technique that attempts to suggest information items (movies, books, music, news, Web pages, images, etc.) that are likely to interest the users. Typically, recommender systems are based on Collaborative Filtering, which is a technique that automatically predicts

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WebMedia'14, November 18–21, 2014, João Pessoa, Brazil.

Copyright 2014 ACM 978-1-4503-3230-9/14/11 ...\$15.00.

<http://dx.doi.org/10.1145/2664551.2664556>.

the interest of an active user by collecting rating information from other similar users or items [12].

The traditional recommendation engines consist in acquiring the preferences of users through profiling techniques based on explicit feedback, implicit feedback and hybrid approaches. The approaches based on explicit information collect explicit data provided by users, such as filling out forms or classification of content. This approach is generally considered more accurate, considering that it is provided directly by users, but require a great effort from them [9]. On the other hand, approaches that capture implicit information indirectly collect user interactions during browsing, such as browsing history and mouse movement. This is a more abundant source of information because they are gathered automatically by the system; however, an analysis of user's behavior must be accomplished to infer positive or negative preferences. Hybrid approach, in turn, is the combination of the two types of feedback to obtain a larger and more accurate amount of information.

In order to obtain such interests, profiling mechanisms have been developed, which consist of acquiring, representing and maintaining pieces of information relevant (and/or irrelevant) to the user. In the particular case of obtaining user's preferences, the three most known techniques are based on explicit feedback, implicit feedback and hybrid approaches. Implicit information is collected indirectly during user navigation with the system while visiting a page, mouse movement and clicks on various links of interest. Regarding explicit feedback, the data is intentionally provided, i.e., the user expresses himself in some direct way (e.g. filling in forms or rating a content). This type of information is considered more reliable, since the user is who provides the topics of interests, but the cost of this procedure is the effort of the individual, who is not always willing to cooperate with the system [1]. Finally, the hybrid approach consists of applying the implicit and explicit feedback together, in order to obtain a greater number of user information [9].

However, the performance can be significantly improved, if ensemble methods are used. An ensemble method combines the predictions of different algorithms (or blending) to obtain a final prediction. The most basic blending method is to compute the final prediction simply as the mean over all the predictions in the ensemble [1]. Better results can be obtained if the final prediction is given by a linear combination of the ensemble predictions. In this case, the combination weights have to be determined by some optimization procedure, in general by regularized linear and logistic regressions. Though, not all available ensemble methods are

practical for large-scale recommender systems because the massive amount of data leads to vast time and memory consumption.

In this paper, we propose a framework to unify different types of feedback from users using an ensemble learning approach. First, each interaction type is used to learn individuals models; and then, the results of each model is combined using a linear regression algorithm based on a Bayesian optimization criterion. We provide an experimental evaluation of our algorithm with the HetRec2011 Last fm 2k [3] dataset, simulating and inferring a number of interaction paradigms: the user’s browsing history and whether he tagged a content or not.

This paper is structured as follows: in Section 2 we depict the related work; Section 3 we give an overview of the notations that we will use during the paper; Section 4 presents the ensemble algorithms developed previously based on heuristics; Section 5 provides a description of the Bayesian Optimization approach, which is explored in this work; in Section 6 we present our proposal in details; Section 7 describes the evaluation executed in the system; and finally, in Section 8 we present the final remarks and future works.

2. RELATED WORK

In this section, we review some work related to our proposal. First, we depict approaches related to multimodal recommender systems, and then, we provide a review of ensemble-based recommender systems.

2.1 Multimodal Interactions

With the increasing number of interactions between users and content, several studies have emerged in order to work with the integration of these interactions, so that more information about the users preferences are gathered by the systems. The work proposed by [14] developed a recommendation system for on line video based on explicit and implicit feedback, plus feedback from relevant information provided by the user. The video used was composed of multimedia content and related information (such as query, title, tags, etc.). The project aimed to combine these types of interactions with the information provided by users in order to generate a more precise rank of relevant items. In order to automatically adjust the system, it was implemented a set of adjustment heuristic given new user interactions.

The SVD++ algorithm proposed by [9] uses explicit and implicit information from users to improve the prediction of ratings. As explicit information, the algorithm uses the ratings assigned by users to items, and as implicit information, it simulates the rental history by considering which items users rated, regardless of how they rated these items. As limitation, the SVD++ algorithm uses a stochastic gradient descent to train the model, which requires the observed ratings from users. Thus, it is impossible to infer preferences for those users who provided only implicit feedback.

In recent research, Domingues et al. [4] developed a multimodal system facing music recommendation, which combines the use (web access) and content (i.e. audio features and textual tags). Part of interactions was done in real time with real users in a commercial music site from the very Long Tail. Combining the data from the system led to better results than content-based systems, leading the system

to have greater user acceptance rate, higher rate of user activity and greater user loyalty and usage.

The approach proposed in this paper differs from the aforementioned works because it adopts a post-processing step to analyze the rankings created separately by different algorithms. The advantage of this approach is that it is easier to extend the model to other types of interactions and recommenders.

2.2 Ensemble Approach

Ensemble is a machine learning approach that uses a combination of similar models in order to improve the results obtained by a single model. In fact, several recent studies, such as [8], demonstrate the effectiveness of an ensemble of several individual and simpler techniques, and show that ensemble-based methods outperform any single, more complex algorithm.

In [1] it is proposed a systematic framework for applying ensemble methods to CF methods. They employ automatic methods for generating an ensemble of collaborative filtering models based on a single collaborative filtering algorithm (homogeneous ensemble). They demonstrated the effectiveness of this framework by applying several ensemble methods to various base CF methods.

In the recent work of [12], they discussed the development of a hybrid multi-strategy book recommendation system using Linked Open Data. Their approach builds on training individual base recommenders and using global popularity scores as generic recommenders. The results of the individual recommenders are combined using ensemble method and rank aggregation. They showed that their approach delivers very good results in different recommendation settings and also allows for incorporating diversity of recommendations. However, their work is limited to the type of interactions chosen by the authors.

Our proposal can be considered an ensemble-based technique, as it combines multiple rankings in a post-processing step. However, our approach differs from the related work in the sense that we analyze multiple interaction paradigms from the user in order to generate a more accurate personalized ranking. Our contribution, thus, can be considered a multimodal recommender system based on multiple user feedback types, but it also uses an ensemble learning technique to generate recommendations.

3. NOTATION

Following the same notation in [11], we use special indexing letters to distinguish users and items: a user is indicated as u and an item is referred as i, j ; and r_{ui} is used to refer to either explicit or implicit feedback from a user u to an item i . In the first case, it is an integer provided by the user indicating how much he liked the content; in the second, it is just a boolean indicating whether the user consumed or visited the content or not. The prediction of the system about the preference of user u to item i is represented by \hat{r}_{ui} , which is a floating point value guessed by the recommender algorithm. The set of pairs (u, i) for which r_{ui} is known are represented by the set $K = \{(u, i) | r_{ui} \text{ is known}\}$.

Additional sets used in this paper are: $N(u)$ to indicate the set of items for which user u provided an implicit feedback, and $\bar{N}(u)$ to indicate the set of items that are unknown to user u . The learning rate of the algorithm is represented

with the variable α and Λ represents constants used for regularization, and are defined by cross-validation.

Particularly in this paper, we define $R(u, tags)$, $R(u, history)$ and $R(u, ratings)$ the rankings generated to user u for the interactions: tags, history navigation and ratings respectively. In addition, concerning these interactions, we define $r_{u,i}^{tags}$, $r_{u,i}^{history}$ and $r_{u,i}^{ratings}$ to represent the scores of pair (u, i) in each ranking.

Thus, the concept of ranking and scores are related to each other: each unimodal algorithm will generate a score (weight) which is a floating point representing how much a user likes an item using a particular interaction. These scores are then sorted in decreasing order forming the ranking of items where the first is the most relevant to that user’s preferences. In this way, for each user and interaction (tagging and navigation), we will have a ranking. For instance, $R(u, tagging)$ contains a list of (u, i) pairs with corresponding scores generated by a unimodal algorithm based on tagging interaction of user u .

4. ENSEMBLE ALGORITHMS BASED ON HEURISTICS

The idea of using multiple interactions from users in recommendation systems by mean of ensembling methods has been explored in two previous work of ours. In spite of their promising results, they were based on a set of heuristics, which works better on a restricted domain.

In the first work [5], we propose a robust framework capable of generating recommendations based on multimodal user interactions, whenever they are available or not. The system consists of a post-processing step which combines rankings generated by different unimodal recommenders exploiting individual interaction types. We used two algorithms: SVD++ and BPR MF which generate rankings based on a variety of feedback types. In this approach, the algorithm prioritizes those items that appear more than once in the $R(u, partial)$ and the items on which the user has assigned tags. This heuristic is supported by the fact the higher the frequency of the item in $R(u, tags)$, $R(u, history)$ and $R(u, ratings)$, the more this item is closer to the user’s preferences (the user has interacted with this content in different ways). In addition, it was found that a higher importance for the parameter β achieved better results; it is because tagging a resource requires more effort from the user than simply accessing an item or giving a rating; consequently, it is inferred that item captured better his attention than others.

We defined in the second work [6], $R(u, tags)$, $R(u, history)$ and $R(u, ratings)$ the rankings generated to user u for the interactions: tagging, history navigation and ratings, respectively. In addition, concerning these interactions, we define $r_{u,i}^{tags}$, $r_{u,i}^{history}$ and $r_{u,i}^{ratings}$ to represent the scores of pair (u, i) in each ranking. After generating the unimodal rankings, our algorithm processes these rankings as illustrated in Algorithm 1. First, a partial ranking $R(u, partial)$ is created containing the (u, i) pairs which occur in all rankings. Then, the average scores of each ranking is computed. Following, for each interaction type and each (u, i) pair in $R(u, partial)$, we test whether the score $s(u, i, \cdot)$ is greater than the corresponding average score. If all scores satisfy the condition, we set the final score $\hat{r}_{u,i}$ for that user and item pair as the highest score among the rankings. Finally, these values

are sorted in descending order resulting in the final ranking which will be recommended at top N .

```

Input:  $R(u, tags), R(u, history), R(u, ratings)$ 
Output: Final Rank  $R'(u, final)$ 
 $R(u, partial) \leftarrow$ 
 $R(u, tags) \cap R(u, history) \cap R(u, ratings)$ 
Compute  $avg_{R(u, tags)}, avg_{R(u, history)}$  and
 $avg_{R(u, ratings)}$ 
for  $(u, i) \in R(u, partial)$  do
  if  $r_{u,i}^{tags} \geq avg_{R(u, tags)}$  &  $r_{u,i}^{history} \geq avg_{R(u, history)}$  &
 $r_{u,i}^{ratings} \geq avg_{R(u, ratings)}$  then
     $\hat{r}_{u,i} \leftarrow \max(r_{u,i}^{tags}, r_{u,i}^{history}, r_{u,i}^{ratings})$ 
  end
  Aggregate  $\hat{r}_{u,i}$  into  $R(u, final)$ 
end
 $R'(u, final) \leftarrow \text{sort\_desc}(R(u, final))$ 

```

Algorithm 1: Heuristic-based Ensemble Algorithm

5. BAYESIAN PERSONALIZED RANKING

As previously exposed, the previous ensembling approaches combine multiple rankings generated by unimodal recommenders using a set of heuristics which were defined based on the considered domains and set of available feedback types. In this way, although we have generated better results when compared to unimodal recommenders, it is difficult to extend the algorithms to different types of interactions, or to be generic enough to any application domain. In this way, we propose an ensembling method whose parameters are learned based on observed data, i.e., the behavior of each user along with his interaction with the system.

In order to make our method generic enough to be used with any type of feedback (including only implicit information), our ensembling framework is learned using a Bayesian Optimization Criterion as defined by [11]. Thus, in this section such procedure is described prior to presenting our proposal.

5.1 BPR Optimization Criterion

The BPR MF approach [11] consists of providing personalized ranking of items to a user according only to implicit feedback (e.g. navigation, clicks, etc.). An important characteristic of this type of feedback is that we only know the positive observations; the non-observed user-item pairs can be either an actual negative feedback or simply the fact that the user does not know about the item’s existence.

In this scenario, Rendle et al. [11] discuss a problem that happens when training an item recommendation model based only on such positive/negative data. Because the observed entries are positive and the rest are negative, the model will be fitted to provide positive scores only for those observed items. The remaining elements, including those which may be of interest to the user, will be classified by the model as negative scores, in which the ranking cannot be optimized as the predictions will be around zero.

Considering this problem, the authors have proposed a generic method for learning models for personalized ranking [11]. Instead of training the model using only the user-item pairs, they also consider the relative order between a pair of items, according to the user’s preferences. It is inferred

that if an item i has been viewed by user u and j has not ($i \in N(u)$ and $j \in \bar{N}(u)$), then $i >_u j$, which means that he prefers i over j . Figure 1 presents an example of this method. It is important to mention that when i and j are unknown to the user, or equivalently, both are known, then it is impossible to infer any conclusion about their relative importance to the user.

| | | | | |
|-------|-------|-------|-------|-------|
| | i_1 | i_2 | i_3 | i_4 |
| u_1 | ? | + | + | ? |
| u_2 | + | ? | ? | + |
| u_3 | + | + | ? | ? |
| u_4 | ? | ? | + | + |
| u_5 | ? | ? | + | ? |

| | | | | |
|-------|-------|-------|-------|-------|
| | i_1 | i_2 | i_3 | i_4 |
| i_1 | ■ | + | + | ? |
| i_2 | - | ■ | ? | - |
| i_3 | - | ? | ■ | - |
| i_4 | ? | + | + | ■ |

$u_1 : i >_u j$

Figure 1: The left-hand side table represents the observed data K . The Rendle et al. approach creates a user-specific pairwise relation $i >_u j$ between two items. In the table on the right-hand side, the plus signal indicates that user u has more interest in item i than j ; the minus signal indicates he prefers item j over i ; and the interrogation mark indicates that no conclusion can be inferred between both items.

To estimate whether a user prefers an item over another, Rendle et al. proposed a Bayesian analysis using the likelihood function for $p(i >_u j | \Theta)$ and the prior probability for the model parameter $p(\Theta)$. The final optimization criterion, BPR-Opt, is defined as:

$$\text{BPR-Opt} := \sum_{(u,i,j) \in D_K} \ln \sigma(\hat{s}_{uij}) - \Lambda_{\Theta} \|\Theta\|^2, \quad (1)$$

where $\hat{s}_{uij} := \hat{r}_{ui} - \hat{r}_{uj}$ and $D_K = \{(u, i, j) | i \in N(u) \ \& \ j \in \bar{N}(u)\}$. The symbol Θ represents the parameters of the model, Λ_{Θ} is a regularization constant, and σ is the logistic function, defined as: $\sigma(x) = 1/(1 + e^{-x})$.

5.2 BPR Learning Algorithm

For learning the model, the authors also proposed a variation of the stochastic gradient descent technique, denominated Learn BPR, which randomly samples from D_K to adjust Θ . Algorithm 2 shows an overview of the algorithm, where α is the learning rate.

Input: D_K
Output: Learned parameters Θ
Initialize Θ with random values
for $count = 1, \dots, \#Iter$ **do**
 draw (u, i, j) from D_K
 $\hat{s}_{uij} \leftarrow \hat{r}_{ui} - \hat{r}_{uj}$
 $\Theta \leftarrow \Theta + \alpha \left(\frac{e^{-\hat{s}_{uij}}}{1 + e^{-\hat{s}_{uij}}} \cdot \frac{\partial}{\partial \Theta} \hat{s}_{uij} - \Lambda_{\Theta} \Theta \right)$
end

Algorithm 2: Learning through Learn BPR.

In this paper, we have defined the BPR approach to consider the prediction rule \hat{r}_{ui} as the simple factorization model as defined in:

$$\hat{r}_{ui} = b_{ui} + p_u^T q_i, \quad (2)$$

where the baseline b_{ui} is defined as $b_{ui} = \mu + b_u + b_i$. In this way, we compute the partial derivatives in relation to \hat{s}_{uij} :

$$\frac{\partial}{\partial \Theta} \hat{s}_{uij} = \begin{cases} 1 & \text{if } \Theta = b_i, \\ -1 & \text{if } \Theta = b_j, \\ q_i - q_j & \text{if } \Theta = p_u, \\ p_u & \text{if } \Theta = q_i, \\ -p_u & \text{if } \Theta = q_j, \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

which is then applied to Algorithm 2 to learn the set of parameters Θ . The use of this prediction rule together with the BPR Learning algorithm is also known as BPR MF [11].

6. PROPOSED METHOD

The previous section described the operation of the BPR algorithm, responsible for rendering implicit feedback and build an accurate representation of the user in order to optimize results in ranks of recommended items. However, in its original form, this model can not process and combine more than one type of interaction. On the other hand, its learning procedure can be used to adjust the parameters of a ensembling learning model. In this way, the combination of assigning tags and user history during navigation, for example, can be made to improve the final recommendation for a user. Consequently, each interaction type is used by a different instance of BPR MF, which, in term, is learned using Algorithm 2. After that, the merging procedure of all rankings is accomplished by the ensembling model, whose related parameters are learned using Algorithm 2 again.

In summary, we adopted the BPR MF and Learn BPR algorithms described in Section 5, which generate rankings based on a variety of feedback types and learn weights to consider different types of interaction. Figure 2 illustrates the overall scheme.

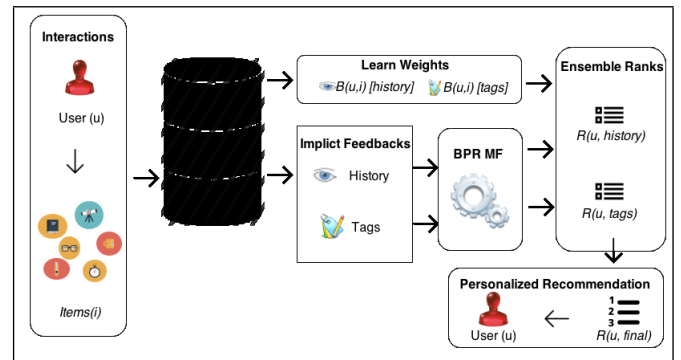


Figure 2: Schematic visualization of the proposed system.

Particularly in this paper, we adopted different types of implicit feedback, although we plan to integrate explicit information in future works. As implicit feedback, we considered two types: i) whether a user assigned a tag or not to an item; and ii) his navigation history. As shown in Figure 2, both implicit feedback types are used by the BPR MF algorithm to generate a personalized ranking for each user.

As illustrated in Figure 2, two rankings will be generated for each user, where each of them was computed based on

a particular feedback. Those rankings are then processed by an ensemble method which will apply a set of heuristics based on the interaction activity of the user. The equation which computes the weight of each pair (u, i) , represented by $r_{u,i}^{final}$, is defined as:

$$r_{u,i}^{final} = \beta_{history} \cdot r_{u,i}^{history} + \beta_{tags} \cdot r_{u,i}^{tags}, \quad (4)$$

where $\beta_{history}$ and β_{tags} are generated weights learned from Learn BPR to weigh each type of interaction.

At this point, it is worth mentioning that our proposal has two phases of training, where firstly the BPR MF parameters of each interaction type are learned from Learn BPR; and secondly, after we have these individual models adjusted for each interaction, we apply the merging procedure (Equation 4), whose parameters $\beta_{history}$ and β_{tags} are learned using another training sample inputted to Learn BPR. In other words, the parameters set θ has different elements depending on the phase of the algorithm: in the beginning, it is set to $\theta = \{b_u, b_i, q_u, q_i\}$, where we have two instances, one for each interaction type; and then, $\theta = \{\beta_{history}, \beta_{tags}\}$ for the ensemble model.

In following subsections, we describe all procedures in more details.

6.1 Learning Weights

Given the input of user interactions, in this step, weights are learned for each of them, through the Learn BPR algorithm seen in Section 5. Using the Algorithm 2, we train individually each instance of BPR MF using particular samples of the dataset consisting of users interactions of that type. This is accomplished by the natural strategy of BPR, where in a particular interaction, we randomly select a pair of items i and j for a user u , where $i \in N(u)$ and $i \in \bar{N}(u)$. After that, the models $r_{u,i}^{history}$ and $r_{u,i}^{tags}$ (both defined by Equation 2) are then merged using Equation 4.

In order to train the ensemble parameters $\beta_{history}$ and β_{tags} , we use the Learn BPR, as previously explained. Indeed, the adjustment is accomplished by the following equation:

$$\beta_\theta \leftarrow \beta_\theta + \alpha \left(\frac{e^{-\hat{s}_{uij}}}{1 + e^{-\hat{s}_{uij}}} \cdot \frac{\partial}{\partial \beta_\theta} \hat{s}_{uij} - \Lambda_{\beta_\theta} \beta_\theta \right), \quad (5)$$

where θ represents the type of interaction (history or tags), α is the learning rate, Λ_{β_θ} the variable convergence and $\hat{s}_{uij} = \hat{s}_{ui} - \hat{s}_{uj}$. Thus, each pair (u, i) of each type of interaction will have an equivalent weight β . In this way, we have:

$$\frac{\partial}{\partial \beta_\theta} \hat{s}_{uij} = \begin{cases} r_{u,i}^{history} - r_{u,j}^{history} & \text{if } \beta_\theta = \beta_{history}, \\ r_{u,i}^{tags} - r_{u,j}^{tags} & \text{if } \beta_\theta = \beta_{tags}, \end{cases} \quad (6)$$

6.2 Ensemble Ranks

The step of combining ranks consists of aggregating the weights of ranks generated by BPR MF for each type of interaction. For each user belonging to the dataset, the scores are computed to each item of their interactions: browsing history and tags. These scores, in turn, are weighted with the ensemble parameters ($\beta_{history}$ and β_{tags}), responsible for giving relevance to the type of interaction the user chooses

in relation to that item. Thus, items in which users had more affinity assigning tags will be more relevant than the other type of interaction and otherwise. Finally, these values are sorted in descending order resulting in the final ranking which will be recommended at top N . This process can be seen in the Algorithm 3.

Input: $interaction_{(history)}, interaction_{(tags)}$
Output: Final Rank $R'(u, final)$

for $u \in users$ **do**
 for $i \in items$ **do**
 Compute $r_{u,i}^{history}, r_{u,i}^{tags}$
 Compute $\beta_{tags}, \beta_{history}$
 Compute $\hat{r}_{u,i}^{final}$
 Aggregate $\hat{r}_{u,i}^{final}$ into $R(u, final)$
 end
end
 $R'(u, final) \leftarrow sort_desc(R(u, final))$

Algorithm 3: Proposed algorithm.

7. EVALUATION

The evaluation presented in this paper aims to compare our approach with the unimodal method described in Section 5 and described in Section 4, which ensemble the ranks through heuristics. The BPR MF implementation used in our work is available in the MyMediaLite library [7]. We generated the recommendations for all users and individual feedback types, and then, implemented as a separate module the ranking combination strategy and the evaluation methodology.

7.1 Dataset

The evaluation of the system was based on the HetRec2011 Last fm 2k [3], consisting of 92,834 user-listened artist relations, 186,479 interactions tags applied by 1,892 users to 17,632 artists. As feedback types, we considered: i) whether a user tagged an item or not; and ii) the history of visited items, which is simulated by boolean values (visited or not) generated by the ratings and tagging activities.

In this paper, we adopted the same methodology used by the research community with regard to recommender systems evaluation. We divide the base into two sets, 80% for training and 20% for testing, where the training set is used to run the isolated algorithms and train matrices p and q ; and test set is used to make *All but One protocol* and the rest serves to predict weights for each pair of algorithms (simulate the real-time interaction from the user).

7.2 Constants

The involved constants used in this evaluation are defined according to Table 1. The details of their utilization can be found in Section 6.

7.3 Methodology

In order to evaluate the proposal in this paper, we adapted the All But One [2] protocol for the construction of the ground truth and 10-fold-cross-validation. Given the data set, randomly we divided into the same 10 subsets and for each sample we use $n - 1$, these subsets of data for training and the rest for testing. The training set t_r was used to test

Table 1: Constants used in the evaluation.

| Constant | Value | Note |
|---------------------|-------------------------------|---|
| α_{tags} | 0.05 | |
| $\alpha_{history}$ | 0.05 | |
| Λ_{tags} | $0.0025 R(i) ^{-\frac{1}{2}}$ | $R(i)$ is the number of items i in set Train. |
| $\Lambda_{history}$ | $0.0025 R(i) ^{-\frac{1}{2}}$ | |

the proposed assembly and test system T_e randomly split an item for each user to create the truth set H . That done, the remaining items form the set of observable O , used to test the unimodal algorithms. To assess the outcomes of the systems we use evaluation metrics Precision and Mean Average Precision (MAP) [13]. Then, we compute Precision and Mean Average Precision as follows:

Precision calculates the percentage of recommended items that are relevant. This metric is calculated by comparing, for each user in the test set T_e , the set of recommendations R that the system makes, given the set of observables O , against the set H :

$$Precision(T_e) = \frac{1}{|T_e|} \sum_{j=1}^{|T_e|} \frac{|R_j \cap H_j|}{|R_j|}. \quad (7)$$

Mean Average Precision computes the precision considering the respective position in the ordered list of recommended items. With this metric, we obtain a single value accuracy score for a set of test users T_e :

$$MAP(T_e) = \frac{1}{|T_e|} \sum_{j=1}^{|T_e|} AveP(R_j, H_j), \quad (8)$$

where the average precision (AveP) is given by

$$AveP(R_j, H_j) = \frac{1}{|H_j|} \sum_{r=1}^{|H_j|} [Prec(R_j, r) \times \delta(R_j(r), H_j)], \quad (9)$$

where $Prec(R_j, r)$ is the precision for all recommended items up to rank r and $\delta(R_j(r), H_j) = 1$, iff the predicted item at rank r is a relevant item ($R_j(r) \in H_j$) or zero otherwise.

In this work we used Precision@ N and MAP@ N , where N took values of 1, 3, 5 and 10 in the ranks returned by the system. For each configuration and measure, the 10-fold values are summarized by using mean and standard deviation. In order to compare the results in statistical form, we apply the two-sided paired t-test with a 95% confidence level [10].

7.4 Results

Tables 2 and 3 show the results of this evaluation, together with the standard deviation. We note that the proposed method achieved statistically better results than the baselines, as proven by the t-student analysis ($p < 0.05$). Figures 3 and 4 illustrate the algorithms' performance in Top@ N vs. MAP and Top@ N vs. Precision graphs.

From Figures 3 and 4 we note that MAP has a tendency for higher values as the number of returned items increases; and precision the opposite. This can be explained because

MAP only considers the relevant items and their positions in the ranking. Thus, as more items are returned, the number of relevant items is also increased. In case of precision, in turn, as it is a set-based measure (the order of items is irrelevant), the more items are filtered to the user, the more false positives may also be returned, affecting, consequently, the precision measure.

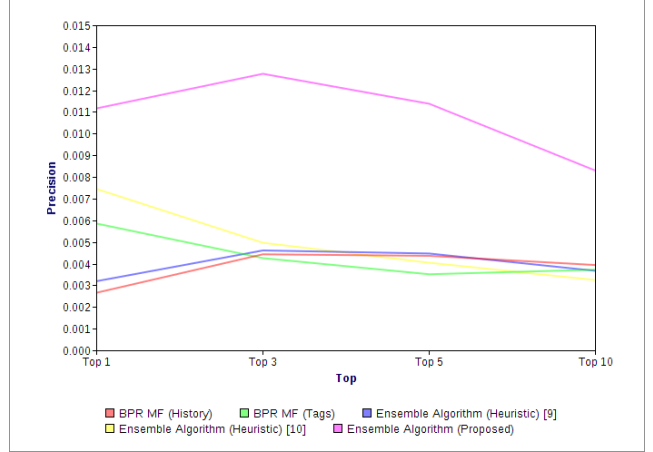


Figure 3: Graph comparing the Precision.

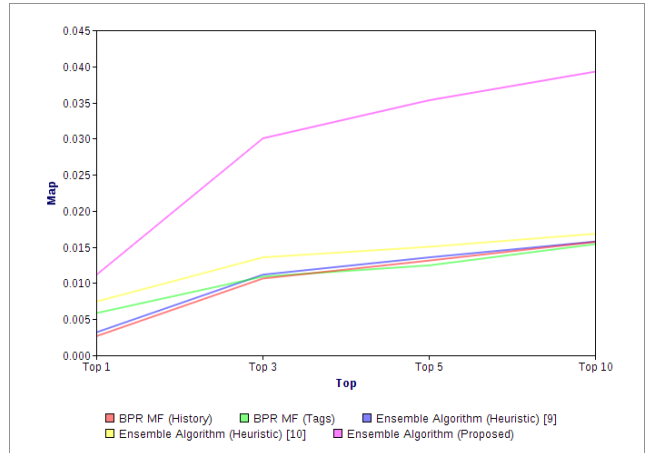


Figure 4: Graph comparing the MAP.

We note that by combining all two types of feedback (tagging and history) using the proposed ensembling algorithm with BPR for learning the weights, we achieved the best results for all top N recommendations. This is because the algorithm is able to learn, according to the input data, the preferences of each user for each type of interaction.

The overall results obtained and described in paper are small because of the evaluation protocol used in the experiments. The All But One hides one item from each user in the test set and considers it as the ground truth. As we are recommending top N items, the precision and MAP will decrease because the system thinks there are N relevant items, although the protocol has set only the hidden item as relevant. In this way, it is important to rely only on the differences among the approaches, and we managed to increase the results of our proposal when compared to the baselines.

Table 2: Comparative Precision table.

| | | Top 1 | Top 3 | Top 5 | Top 10 |
|--------------------------------------|--------------------|-----------------|-----------------|-----------------|-----------------|
| BPR MF (History) | Precision | 0.002658 | 0.00443 | 0.004359 | 0.003934 |
| | Standard deviation | 0.000012 | 0.000037 | 0.000108 | 0.000055 |
| BPR MF (Tags) | Precision | 0.005848 | 0.004253 | 0.003509 | 0.003721 |
| | Standard deviation | 0.000142 | 0.000122 | 0.000087 | 0.000105 |
| Ensemble Algorithm (Heuristic) [5] | Precision | 0.00319 | 0.004607 | 0.004466 | 0.003668 |
| | Standard deviation | 0.000032 | 0.000187 | 0.000098 | 0.000131 |
| Ensemble Algorithm (Heuristic) [6] | Precision | 0.007443 | 0.004962 | 0.004041 | 0.003243 |
| | Standard deviation | 0.000217 | 0.000320 | 0.000063 | 0.000081 |
| Ensemble Algorithm (Proposed) | Precision | 0.011164 | 0.012759 | 0.011377 | 0.008293 |
| | Standard deviation | 0.000098 | 0.000123 | 0.000027 | 0.000032 |

Table 3: Comparative MAP table.

| | | Top 1 | Top 3 | Top 5 | Top 10 |
|--------------------------------------|--------------------|-----------------|-----------------|-----------------|-----------------|
| BPR MF (History) | MAP | 0.002658 | 0.010633 | 0.013114 | 0.015704 |
| | Standard deviation | 0.000121 | 0.000064 | 0.000169 | 0.000201 |
| BPR MF (Tags) | MAP | 0.005848 | 0.010898 | 0.012449 | 0.015395 |
| | Standard deviation | 0.000143 | 0.000529 | 0.000287 | 0.000458 |
| Ensemble Algorithm (Heuristic) [5] | MAP | 0.00319 | 0.011164 | 0.013557 | 0.015749 |
| | Standard deviation | 0.000245 | 0.000318 | 0.000266 | 0.000374 |
| Ensemble Algorithm (Heuristic) [6] | MAP | 0.007443 | 0.013557 | 0.015019 | 0.016827 |
| | Standard deviation | 0.000497 | 0.000123 | 0.000075 | 0.000192 |
| Ensemble Algorithm (Proposed) | MAP | 0.011164 | 0.030037 | 0.035309 | 0.039275 |
| | Standard deviation | 0.000428 | 0.000137 | 0.000065 | 0.000236 |

8. FINAL REMARKS

This paper proposed an ensembling approach to unify different types of feedback from users when consuming content in order to provide better recommendations. The advantage is that more information about the interests of the user can be obtained when analyzing multimodal interactions. In contrast to existing approaches which are limited to one or a small subset of user feedback, resulting in inaccurate representation of users' preferences, the proposed model incorporates the feature of using various types of interactions, but still taking advantage of state-of-the-art algorithms which are based on unimodal feedback.

We depicted an evaluation of the proposed method, comparing it against four baselines approaches. The experiments were executed with the HetRec2011 Last fm 2k dataset, and the results show the effectiveness of combining various types of interactions in a single model for recommendation using ensemble learning. In fact, our learning procedure is accomplished by means of BPR, which is a generic framework that allows fast optimization of ranks by analyzing a triple of user, observed item and unknown item. In this paper, in particular, we explore this idea to combine different types of implicit feedback.

In future works, we intend to consider other types of interaction and context information of users and items, and also other recommenders with better accuracy for a single feedback type. We intend to test our new algorithm in databases that contain three types of interactions to make a comparative study of results.

9. REFERENCES

- [1] A. Bar, L. Rokach, G. Shani, B. Shapira, and A. Schclar. Improving simple collaborative filtering models using ensemble methods. In Z.-H. Zhou, F. Roli, and J. Kittler, editors, *Multiple Classifier Systems*, volume 7872 of *Lecture Notes in Computer Science*, pages 1–12. Springer Berlin Heidelberg, 2013.
- [2] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, pages 43–52, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.
- [3] I. Cantador, P. Brusilovsky, and T. Kuflik. 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In *Proceedings of the 5th ACM conference on Recommender systems*, RecSys 2011, New York, NY, USA, 2011. ACM.
- [4] M. Domingues, F. Gouyon, A. Jorge, J. Leal, J. Vinagre, L. Lemos, and M. Sordo. Combining usage and content in an online recommendation system for music in the long tail. *International Journal of Multimedia Information Retrieval*, 2(1):3–13, 2013.
- [5] A. Fortes, M. Domingues, S. Rezende, and M. Manzato. Improving personalized ranking in recommender systems with multimodal interactions. *Web Conference Intelligence - WIC*, 2014.
- [6] A. Fortes and M. Manzato. Multimodal interactions in recommender systems: An ensembling approach. *Brazilian Conference on Intelligent Systems - BRACIS*, 2014.
- [7] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Mymedialite: A free recommender system library. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 305–308, New York, NY, USA, 2011. ACM.

- [8] M. Jahrer, A. Töschler, and R. Legenstein. Combining predictions for accurate recommender systems. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 693–702, New York, NY, USA, 2010. ACM.
- [9] Y. Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 426–434, New York, NY, USA, 2008. ACM.
- [10] T. M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.
- [11] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *CoRR*, abs/1205.2618, 2012.
- [12] P. Ristoski, E. L. Mencia, and H. Paulheim. A hybrid multi-strategy recommender system using linked open data. *ESWC*, 2014.
- [13] E. M. Voorhees and D. K. Harman. *TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing)*. The MIT Press, 2005.
- [14] B. Yang, T. Mei, X.-S. Hua, L. Yang, S.-Q. Yang, and M. Li. Online video recommendation based on multimodal fusion and relevance feedback. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, CIVR '07, pages 73–80, New York, NY, USA, 2007. ACM.