Evaluating Multiple User Interactions for Ranking Personalization Using Ensemble Methods

Frederico Araújo Durão, Bruno Souza Cabral
Department of Computer Science
Federal University of Bahia
Salvador, BA – Brazil
fdurao@ufba.br, bruno.cabral@ufba.br

Renato D. Beltrão, Marcelo G. Manzato
Mathematics and Computing Institute
University of São Paulo
São Carlos, SP – Brazil
rdompieri@usp.br, mmanzato@icmc.usp.br

***Abstract*—The variety of interaction paradigms on the Web, such as clicking, commenting or rating are important sources that help recommender systems to gather accurate information about users' preferences. Ensemble methods can be used to combine all these pieces of information in a post-processing step to generate recommendations that are more relevant. In this paper, we review the application of existing ensemble methods to improve ranking recommendations in the multimodal interactions context. We compared four ensemble strategies, ranging from simple to complex algorithms including Gradient Descent and Genetic Algorithm to find optimal weights. The evaluation using the HetRec 2011 MovieLens $2k$ dataset with three different types of interactions shows that a considerable 7% improvement in the Mean Average Precision can be achieved using ensembles when compared to the most performant single interaction.**

***Index Terms*—recommender system, multimodal, user interaction**

I. Introduction

According to [1], from 2005 to 2020, the information in the digital universe will grow by a factor of 300, from 130 exabytes to 40,000 exabytes, or 40 trillion gigabytes (more than 5,200 gigabytes for every man, woman, and child in 2020). It is simply not possible to grasp even a small percentage of it in a single lifetime, there is too much information to process and to choose. The expression *Information Overload* was introduced to describe the sensation of fatigue and distress that follows the cognitive surplus required to handle the volume of information we have to deal with everyday [2].

Recommender Systems (RS) have emerged in response to the information overload problem in order to support users during content consumption decisions. They learn the users' interests using their past interactions (ratings, votes, ranked lists, mouse clicks, page views, product purchases, etc.) and suggest products that are likely to be appreciable. In order to obtain users' interests, three different forms can be used: explicit feedback; implicit feedback and hybrid approaches. Implicit feedback is the kind of information collected indirectly, such as mouse movements or clicks. In explicit feedback, the preferences are intentionally provided by the user, such as a "like" option or a rating. This type of information is considered more reliable, since the user is the one who exposes his interests, rather than being inferred. The problem is that it requires an additional effort from the user to intentionally provide the feedback, who is not always willing to cooperate with the system [3]. Finally, the hybrid approach consists of applying the implicit and explicit feedback together to obtain more information about his preferences [4].

10.18293/SEKE2018-0112

Despite the variety of ways to collect users' preferences, actual recommender algorithms are modeled based on a single or a few types of interactions [5]. However, the accuracy can be improved if the system utilizes all available information. An approach for generically handling multimodal interactions is with ensemble methods. An ensemble method combines the predictions of different algorithms, or the same algorithm with different parameters to obtain a final prediction. Ensemble methods were the top performing solution in the Netflix Prize contest [6].

The most simple ensemble method is to compute the final prediction as the mean over all the predictions [3]. 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 such as regularized linear and logistic regressions. However, not all available ensemble methods are practical for large-scale recommender systems because the massive amount of data demands vast amount of time and memory consumption.

In this paper, we analyze four ensemble strategies ranging from simple rank list merging to advanced strategies using Gradient Descent and Genetic Algorithm to find optimal weights to unify different types of feedback. We provide an experimental evaluation of those strategies with the HetRec2011 MovieLens $2k$ [7] dataset, simulating and inferring three classical users' interactions: tagging, rating and browsing history.

This paper is structured as follows: Section II depicts the related work; Section III details the evaluated ensemble framework and strategies; Section IV presents the evaluation and validation of the approach with HetRec dataset with 800,000 ratings, along with an analysis of the performance of the four strategies; and finally Section V discusses the final remarks and future works.

II. Related Work

Recommender systems can be extended in several ways aiming at improving the understanding of users and items, incorporating new types of interaction in the recommendation process and making the combination of them. One of these improvements is the support for multi-criteria interactions, so as to provide greater flexibility and less obtrusive types of recommendations [8]. In this context, with more studies in the area of recommender systems, various algorithms enabled the usage of more than one type of user interaction.

These studies resulted in works such as Johansson [9], responsible for developing the MADFILM, a movie recommendation system that addresses the integration of prediction and organization of content, through explicit and implicit user's feedback. The work proposed by [10] developed a recommendation system for online video based on explicit and implicit feedback, plus feedback from relevant information provided by the user. The used video 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 heuristics given new user interactions.

The SVD++ algorithm proposed by [11] 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. However, it use 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.

Ensemble is a machine learning approach that uses a combination of similar models in order to improve the results obtained by a single model, and can be used to combine multiple interactions. In fact, several recent studies, such as [12], demonstrate the effectiveness of an ensemble of several individual and simpler techniques, and show that ensemble-based methods outperform any single, more complex algorithm. Most of the related works in the literature point out that ensemble learning has been used in recommender system as a way to combine the prediction of multiple algorithms (heterogeneous ensemble) to create a more accurate rank [12], in a technique known as *blending*. Furthermore, they have been used with a single collaborative filtering algorithm (single-model or homogeneous ensemble), with methods such as *Bagging* and *Boosting* [3].

Cabral et al. [13] proposed three ensemble strategies that combine predictions from a recommender trained with distinct item metadata into a unified rank of recommended items. In comparison, da Costa at al. [14], proposed a similar ensemble strategy based on machine learning in order to combine different types of interactions generated by multiple recommenders. Those strategies differ from the aforementioned works because they adopt a post-processing step to analyze the rankings created separately by different algorithms. The advantage of this approach is that it does not require the algorithm to be modified, or to be trained multiple times with the same dataset, and therefore, it is easier to extend the models to other types of interactions and recommenders. We implemented all strategies in a public available repository and evaluated three types of interactions (Ratings, Tags and Visualized Items) using the HetRec dataset [7].

III. Ensemble Models

In this section, we describe four ensemble strategies used in this work to combine multimodal interactions: **Most Pleasure**, the simplest ensemble strategy, combines predictions based on score; **Best of All strategy**, determines a preferred metadata for a user and uses it to create the ensemble; **Weighting strategy**, uses multiple metadata and weighs them with a Genetic Algorithm, optimizing for maximum Mean Average Precision (MAP); **BPR Learning Strategy** [15], which uses the Learn BPR to learn the optimal weights, optimizing for the Area under the ROC curve (AUC) .

A. *Most Pleasure Strategy*

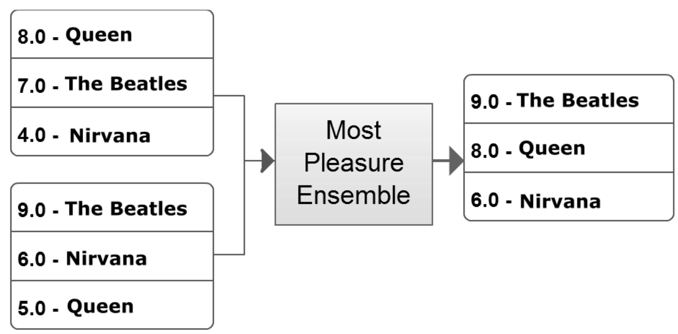

Fig. 1. Most Pleasure Strategy.

The *Most Pleasure* strategy is a classical aggregation method, often used for combining individual ratings for group rating [16]. It takes the maximum of individual ratings for a specific item and creates a unified rank. Figure 1 illustrates the *Most Pleasure* strategy, in which the output comprehends a ranked list of artists with highest ratings from two distinct input sets. It only needs the generated prediction set as an input, composed of the predictions from the recommender algorithm trained with one of the item's metadata. For each user, a new prediction is created, selecting the highest score of an item among all the individually-trained algorithms.

The idea behind this strategy is that differently trained algorithms have a distinct knowledge about the user's preferences, and the predicted score can be considered an indicator of the algorithm's confidence. Consequently, the created ensemble is a list of items in which the distinct algorithms have more confidence to recommend.

https://github.com/wendelad/RecSys

B. Best of All Strategy

Differently from *Most Pleasure*, *Best of All* strategy assumes that different types of metadata can affect users differently. It considers the recommendation algorithm that provides the best results for a specific user (as illustrated by Figure 2).

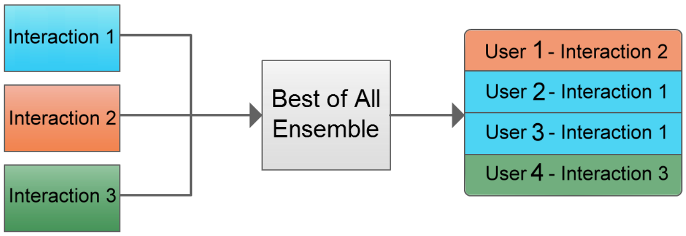

Fig. 2. Best of All Strategy.

The *Best of All* Strategy requires as input: i) the recommendation algorithm, ii) a training dataset, iii) a probe dataset, and iv) the set of item's metadata. Unlike the *Most Pleasure* strategy, this one requires a probe run to determine which is the best performing algorithm. Therefore, the dataset is divided in training and probe. The recommender algorithm is firstly trained using each of item metadata individually. Then, for each user, a probe run is made to determine the metadata with the highest performance (in terms of MAP). Finally, the recommender algorithms are retrained using all data (including the probe set), and the final ensemble is the result of the combination of predictions using, for each user, the prediction from the algorithm with the highest performance in the probe test.

The idea behind this strategy is that a single metadatum can greatly influence the user's preferences, and this should be used for future predictions. For instance, if a User A enjoys music from a particular genre such as "pop", and other User B enjoys music of some specific performer such as "Metallica", the ensemble will contain predictions from the recommendation algorithm trained with both: the genre metadatum for User A, i.e. "pop", and a performer metadata for user B, i.e. "Metallica".

C. GA Weighting Strategy

One drawback of the *Best of All* strategy is that it considers that only one type of metadata influences the user preference. The *GA Weighting* strategy assumes that the interests of a user may be influenced by more than one metadatum, and with different levels. It considers all available metadata, assigning different weights for each prediction as shown in Figure 3.

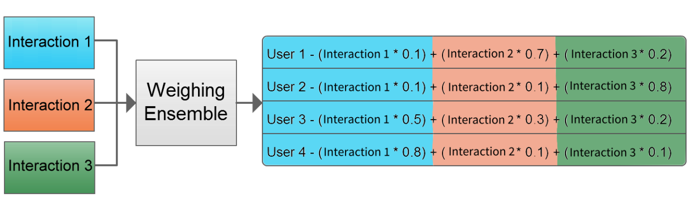

Fig. 3. Weighting Strategy.

Similarly to the previous strategy, it requires as an input the i) recommendation algorithm, ii) a training and probe dataset, and iii) the set of item metadata. After training the algorithm using each of item metadata individually, a probe run is needed; however, the objective is to determine the optimal weights for each user. This is an optimization problem that was solved using Genetic Algorithm (GA).

The probe part consists of running the GA to find out the optimal weights. It was implemented using the GA Framework proposed by Newcombe [17], where the weights are the chromosomes, and the fitness function is the MAP score against the probe dataset. Other GA characteristics includes the use of 5% of Elitism, Double Point crossing-over, and Binary Mutations. Finally, the algorithms are retrained using all data (including the probe set), and the final ensemble uses, as the item score, the sum of individual predictions multiplied by the weights found in the probe phase and divided by the total number of metadata.

The idea behind it is that the different types of interactions influence differently the user preference. Still in the context of music, let us consider that a User A enjoys songs of a specific set of genres regardless of the performer and a User B that does not care about music genre or country of production. For the User A, the ensemble should give a higher weight for the music genre, and a lower weight for the production country. In contrast, to the User B, the ensemble should equally distribute the weights between those metadata.

D. BPR Learning Strategy

In order to combine the output generated by each recommendation technique trained with a different kind of interaction, this ensemble strategy is based on a machine learning algorithm [14].

Firstly, it extracts information about users' interactions from the database, such as sets of tags, ratings and browsing history. With these interactions available, it runs the recommendation algorithms, which receive as input the users' interactions. In this step, each algorithm runs with a particular set of feedback, resulting in a feedback-specific personalized ranking (individual ranking) for each user. Thus, a feedback-specific ranking contains the items and their associated scores, which represent how much a user likes an item described by the considered set of attributes. The final step consists of combining all considered rankings into a final list of recommendations. To do that, it assigns weights according to the relevance of each type/set of attributes. This combination is performed according to a linear function, represented by $\hat{r}_{u,i}^{final}$:

$$\hat{r}_{ui}^{final} = \beta_a r_{ui}^a + \beta_b r_{ui}^b + ... + \beta_n . r_{ui}^n \tag{1}$$

where r_{ui}^a, r_{ui}^b, ..., r_{ui}^n indicate the scores computed previously by each individual recommendation algorithm for a (u, i) pair, and β_a, β_b, ..., β_n are the weights of each individual score for the final prediction, learned using Learn BPR algorithm [18]. This is possible because of the natural strategy of BPR, which in a each interaction, select randomly a couple of items i and j for a user u, a known item i and one unknown item j.

Finally, the algorithm predicts scores for items not seen by each user and sorted these scores in descending order resulting in the final ranking, which will be recommended in a top N ranking list. The underlying characteristic of this algorithm is the ability to learn the users' preferences to employ this information to match the recommendations generated individually for each type of interaction.

IV. Evaluation

The evaluation consists in comparing the four ensemble strategies as presented in Section 3, using a standard dataset available in the literature. Three different interactions, history(watched movies), tags and ratings were trained individually and combined using the ensemble techniques. The combined results and individual interactions were evaluated to check the contribution of each aspect to the final recommendation improvement.

A. Dataset

In order to evaluate the performance of the ensemble strategies, we used the HetRec MovieLens $2k$ dataset [7]. MovieLens $2k$ consists of 800,000 ratings, 10,000 interactions tags applied to 2,113 users and 10,197 movies. As explicit information, we used the ratings that users assigned to items, and as implicit interaction, 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 a classical methodology used by the research community with regard to recommender systems evaluation [8]. We divide the full dataset into two sets, 80% for training and 20% for testing. The training set is used to run the isolated algorithms and predict weights for each pair of algorithms (simulate the real-time interaction from the user); and test set is used with the *All but One protocol* to evaluate the approaches.

B. Experimental Setup and Evaluation Metrics

In this evaluation we use the All But One [19] protocol for the construction of the ground truth and the 10-fold-cross-validation. We randomly divided the dataset into 10 disjoint subsets of equal size and for each sample we use $n-1$ of data for training and the rest for testing. The training set t_r was used to train the proposed ensemble and test system T_e randomly split an item for each user to create the truth set H. The remaining items form the set of observable O is used to test the unimodal algorithms. We also evaluated using the standard protocol, where all items are considered. To assess the outcomes of the systems we use evaluation metrics Precision and Mean Average Precision (MAP) [20]. 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:

$$Precision(T_e) = \frac{1}{|T_e|}\sum_{j=1}^{|T_e|}\frac{|R_j \cap H_j|}{|R_j|} \tag{2}$$

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:

$$MAP(T_e) = \frac{1}{|T_e|}\sum_{j=1}^{|T_e|} AveP(R_j, H_j) \tag{3}$$

where the average precision (AveP) is given by

$$AveP(R_j, H_j) = \frac{1}{|H_j|}\sum_{r=1}^{|H_j|}[Prec(R_j, r) \times \delta(R_j(r), H_j)] \tag{4}$$

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

The *GA Weighting* Strategy, which utilizes a Genetic Algorithm (GA), uses a population of size 40 with 90 generations, a crossover probability of 80% and a mutation probability of 8%. This is a small number of generations, and usually a much higher number of generations is used for convergence; however, due to the size of our dataset, we traded precision for speed.

In this work we used Precision@N and MAP@N, where N took values of $1, 3, 5$ and 10 in the rankings 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 [21].

C. Methodology

For implicit data interactions (history and tags), we used the BPR-MF implementation available in the MyMediaLite library [22]. It is an implementation of the Bayesian Personalized Ranking (BPR) [23], a generic framework for optimizing different kinds of models based on training data containing only implicit feedback information. For explicit interactions (ratings), we used SVD++ [4], also from MyMediaLite library. All four ensemble strategies were implemented using MyMediaLite library and are publicly available.

All the runtime evaluations were executed in the same machine, a Core i7-2670QM with 8GB of RAM, with the .NET 4.5 framework with all avaliable patches applied. The result is the average of 10 runs.

D. Results

Table 1 and Table 2 show the results of this evaluation for single interactions and ensembles. We compared our results to *tags*, the best performing interaction. As seen, the *BPR Learning* strategy achieved statistically better results than the

https://github.com/wendelad/RecSys

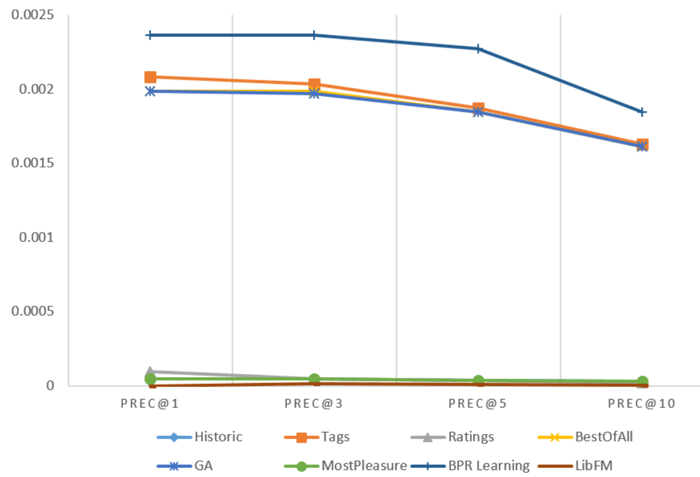

Fig. 4. Precision@1, 3, 5, 10 using All But One Protocol.

baseline, as proven by the t-student analysis (with $p < 0.05$) in Table 3. Figure 4 illustrates the algorithms' precision@1, 3, 5 and 10 using the All But One Protocol.

TABLE I
ALGORITHMS' PERFORMANCE IN PRECISION@1, 3, 5 AND 10.

	Prec@1	Prec@3	Prec@5	Prec@10
History	0.000047	0.000047	0.000037	0.000033
Tags	0.002082	0.002035	0.001874	0.001628
Ratings	0.000094	0.000047	0.000037	0.000018
BestOfAll	0.001988	0.001988	0.001845	0.001614
GA	0.001988	0.001971	0.001845	0.001614
MostPleasure	0.000047	0.000047	0.000037	0.000033
BPR Learning	**0.002366**	**0.002366**	**0.002271**	**0.001845**
Ensemble improvement	12.0 %	12.1 %	21.1 %	13.3 %

TABLE II
ALGORITHMS' PERFORMANCE IN MAP@1,10 AND MAP UNDER STANDARD PROTOCOL.

	MAP@5	MAP@10	MAP(Std)
History	0.000104	0.000120	0.000226
Tags	0.004569	0.005456	0.004729
Ratings	0.000119	0.000119	0.000047
BestOfAll	0.004458	0.005345	0.004956
GA	0.004441	0.005334	0.004999
MostPleasure	0.000104	0.000120	0.000239
BPR Learning	**0.005229**	**0.006044**	**0.005075**
Ensemble improvement	14.4 % %	10.7 %	7.2 %

The results from Table 1 and Table 2 indicate that in some cases, ensembles got significantly higher scores than single interactions. The improvement level was between 7.2% and 21.1% compared to the best performing interaction. These improvements were significant as increasing the MAP and precision is a challenge, and every increment in MAP is hard to achieve. Surprisingly, the *tags* interaction achieved higher scores compared to other single interactions. This is an interesting result, because *tags* contain a more diverse set of information, which probably simulate an ensemble. The *BPR Learning* strategy was optimal for all given scenarios since it uses all interactions to make predictions, and it assigns different weights to the most relevant metadata according to the taste of each individual user. On the other hand, the *MostPleasure* strategy achieved the lowest performance among the ensemble strategies.

TABLE III
T-TEST COMPARING MAP@5 USING BPR LEARNING WITH TAGS.

	BPR Learning	Tags
Mean	0.005115	0.004569
Variance	3.16E-07	1.07E-07
Observations	10	10
df	14	
t Stat	-2.65099	
P(T<=t) one-tail	0.009496	
t Critical one-tail	1.76131	
P(T<=t) two-tail	0.018992	
t Critical two-tail	2.144787	

The *GA Weighting* and *Best of All* strategies obtained a good performance, close to the best performing interaction, except in MAP with the standard protocol, where it archived a better result. The *Best of All* strategy is simple to implement and does not require weight optimization, an expensive step in the process required for *BPR Learning* and *GA Weighting*. Alternatively, *GA Weighting* does requires a weight optimization step, but as it uses a Genetic Algorithm, one can manually set the parameters and tradeoff speed or performance.

TABLE IV
COMPARISON OF ENSEMBLE ALGORITMS RUNTIME IN MINUTES AND IMPROVMENT (IN MAP) OVER THE BEST PERFORMING INTERACTION (TAGS).

Ensemble	Probe Run	Time(min)	Improvement
MostPleasure	No	¡ 0.1	-94%
BestOfAll	Yes	¡ 0.1	4.7%
GA	Yes	43.2	5.6%
BPR Learning	Yes	52.1	7.2%

Table IV lists the ensemble strategies runtime and the need for a probe run. *Most Pleasure* has the advantage of not requiring a probe run, but in our evaluation achieved the worst result of all compared strategies with a 94 % lower performance. *BPR Learning* and *GA Weighting* achieved 5.6 % and 7.2 % MAP improvement respectively with a slight runtime advantage for *GA Weighting*. *Best of All* achieved a good performance improvement compared to the best performing single interaction with the advantage of being fast to compute.

The overall absolute scores obtained and described in this paper are small because of the Sparsity and evaluation protocol used in the experiments. The All But One protocol 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 considers there are N relevant items, although the protocol has set only the hidden item as relevant. The high sparsity stands as another challenge, as many movies were not rated, only tagged. In this case, the rating prediction cannot be made. Another issue is that the rating rank is build using the rating predictions in a decreasing order from the SVD++ algorithm and the dataset can contain items with a low score, lowering the metrics related to this interaction as the test dataset is generated randomly. 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.

Finally, we conclude that ensemble algorithms significantly improved the recommender prediction performance, with the

BPR Learning strategy standing out with higher performance improvement on most of the scenarios followed by *GA Weighting* strategy with a lower performance but with a slight smaller runtime and the *Best of All* strategy, whose the highlight is being almost instantaneous to compute.

V. Conclusion

In this paper we evaluated four ensemble strategies 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. All strategies evaluated do not require modification of the recommender algorithm, namely *Most Pleasure*, *Best of All*, *Genetic Algorithm Weighting* and *BPR Learning*. The considered recommender algorithms did not take advantage of multiple types of interactions and the evaluated ensemble algorithms were able to enable those recommenders to take advantage of all interactions. *Most Pleasure*, the simplest strategy, consisted of combining the predictions based on score. *Best of All* determined a single metadata that was more preferred for a user, and the Weighting strategy uses multiple interactions and weights them with a Genetic Algorithm that optimizes the MAP and finally, *BPR Learning* uses LearnBPR to optimize the weights related to AUC. Results from the experiments show the effectiveness of combining various types of interactions in a single model for recommendation using ensemble learning. Our evaluation showed a considerable MAP improvement between 10.7% and 21.1% when using the ensemble algorithms, with the *BPR Learning* producing the best recommendation for the majority of scenarios. These encouraging results indicate that ensemble algorithms can be used to enhance the recommender algorithms with multiple interactions.

As future work, we plan to implement more complex ensemble strategies and evaluate the algorithms with a higher number of metadata in order to verify whether multimodal information can generate better recommendations. In order to do so, it will be necessary to find a more extensive dataset and to evaluate the algorithms runtime performance with this increased work.

Acknowledgment

The authors would like to thank the financial support from FAPESP (proc. number 2016/2028-6).

References

[1] I. Management Association, *Big Data: Concepts, Methodologies, Tools, and Applications: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2016. [Online]. Available: https://books.google.com.br/books?id=BKEoDAAAQBAJ

[2] C. Musto, "Enhanced vector space models for content-based recommender systems," in *Proceedings of the Fourth ACM Conference on Recommender Systems*, ser. RecSys '10. New York, NY, USA: ACM, 2010, pp. 361–364. [Online]. Available: http://doi.acm.org/10.1145/1864708.1864791

[3] A. Bar, L. Rokach, G. Shani, B. Shapira, and A. Schclar, "Improving simple collaborative filtering models using ensemble methods," in *Multiple Classifier Systems*. Springer, 2013, pp. 1–12.

[4] 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*, ser. KDD '08. New York, NY, USA: ACM, 2008, pp. 426–434. [Online]. Available: http://doi.acm.org/10.1145/1401890.1401944

[5] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 1, no. 1, pp. 67–82, Apr 1997.

[6] A. Töscher, M. Jahrer, and R. M. Bell, "The bigchaos solution to the netflix grand prize," *Netflix prize documentation*, 2009.

[7] 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*, ser. RecSys 2011. New York, NY, USA: ACM, 2011.

[8] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*. Springer, 2011, pp. 1–35.

[9] P. Johansson, "Madfilm - a multimodal approach to handle search and organization in a movie recommendation system," *Proceedings of the 1st Nordic Symposium on Multimodal Communication*, pp. 53–65, 2003.

[10] 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*, ser. CIVR '07. New York, NY, USA: ACM, 2007, pp. 73–80.

[11] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *IEEE Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[12] M. Jahrer, A. Töscher, and R. Legenstein, "Combining predictions for accurate recommender systems," ser. KDD '10. New York, NY, USA: ACM, 2010, pp. 693–702. [Online]. Available: http://doi.acm.org/10.1145/1835804.1835893

[13] B. Souza Cabral, R. Dompieri Beltrao, M. Garcia Manzato, and F. Araújo Durão, "Combining multiple metadata types in movies recommendation using ensemble algorithms," in *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web*, ser. WebMedia '14. New York, NY, USA: ACM, 2014, pp. 231–238. [Online]. Available: http://doi.acm.org/10.1145/2664551.2664569

[14] A. da Costa Fortes and M. G. Manzato, "Ensemble learning in recommender systems: Combining multiple user interactions for ranking personalization," in *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web*, ser. WebMedia '14. New York, NY, USA: ACM, 2014, pp. 47–54.

[15] A. F. d. Costa and M. G. Manzato, "Multimodal interactions in recommender systems: An ensembling approach," in *2014 Brazilian Conference on Intelligent Systems*, Oct 2014, pp. 67–72.

[16] J. Masthoff, "Group recommender systems: Combining individual models," in *Recommender Systems Handbook*. Springer, 2011, pp. 677–702.

[17] J. Newcombe, "Intelligent radio: An evolutionary approach to general coverage radio receiver control," Master's thesis, DeMontfort University, UK, 2013.

[18] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, ser. UAI '09. Arlington, Virginia, United States: AUAI Press, 2009, pp. 452–461.

[19] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," ser. UAI'98, San Francisco, CA, USA, 1998, pp. 43–52. [Online]. Available: http://dl.acm.org/citation.cfm?id=2074094.2074100

[20] E. M. Voorhees and D. K. Harman, *TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing)*. The MIT Press, 2005.

[21] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997.

[22] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "MyMediaLite: A free recommender system library," in *Proceedings of the 5th ACM Conference on Recommender Systems*, ser. RecSys '11, New York, NY, USA, 2011, pp. 305–308.

[23] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme, "Learning attribute-to-feature mappings for cold-start recommendations," in *2010 IEEE 10th International Conference on Data Mining (ICDM)*, dec. 2010, pp. 176–185.