Music Recommendations for Groups of Users

Pedro Dias
Dep. Computer Science
Universidade Nova Lisboa
Portugal
p.dias@campus.fct.unl.pt

João Magalhães
Dep. Computer Science
Universidade Nova Lisboa
Portugal
jm.magalhaes@fct.unl.pt

ABSTRACT

This paper presents an algorithm capable of providing meaningful recommendations to small sets of users. We consider not only rating patterns, bias tendencies, and temporal fluctuations, but also group-leaders. The approach here presented intends to bring a fresh new look over group recommendations, making use of latent factor space to identify groups and make recommendations. Although these recommendations are oriented towards a few users, the preferences of their respective group leaders (users that better represent the group) are also taken into account to diversify and smooth these recommendations. In contrast to the majority of group recommender systems described in literature, our system employs a collaborative filtering approach based on latent factor space instead of content-based or ratings merging approaches.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]

Keywords

Collaborative-filtering, groups, recommendations.

1. INTRODUCTION

Recommender systems emerged with the purpose of providing personalized and meaningful content recommendations based on user preferences and usage history. Relying on the closest friends, family members or anyone else with whom one shares similarities to give trustworthy and useful advices has always been a characteristic of human behaviour, and different opinions weigh differently when it comes to making the final choice. The limitation on receiving good opinions from other people starts with the fact that, usually, one does not have many trustworthy or like-minded people to rely on for getting advice, and those few people have very limited knowledge, considering everything that exists and can be recommended, on a global scale point of view.

ImmersiveMe'13, October 22, 2013, Barcelona, Spain.

http://dx.doi.org/10.1145/2512142.2512151.

To explore the large number of ratings available on many online applications, we followed a collaborative filtering approach to inter-relate and mine the relations between users and their preferences.

Within collaborative filtering techniques, latent factor approaches are very popular. The purpose of latent factor approaches to recommender systems is to map both users and products onto the same latent factor space, representing these as vectors with k dimensions:

$$\vec{p_u} = (u_1, u_2, \cdots, u_k), \vec{q_i} = (i_1, i_2, \cdots, i_k) \quad (1)$$

Here, p_u is the user u factors vector, q_i is the product i factors vector and k is the number of latent factors (dimensions) upon which each user u and each product i are represented. By representing users and products in such way, one can evaluate the extent to which users and products share common characteristics by comparing their k factors against each other. The principle underlying this approach is that both users and products can be represented under a common reduced dimensionality space of latent factors that are inferred from the data and explain the rating patterns. Our algorithm operates exclusively in the latent-factor space.

In the context of group recommendation, where there is more than one user to please, recommendations must be provided in a different way so that the whole group of users is satisfied. By operating in the latent-factor space one can easily relate different users. Moreover, by clustering this space we obtain a set of interest-groups to which users belong to. The leaders of these interest-groups are later used to broaden recommendations and cover products that satisfy all users.

This paper is organized as follows: section 3 describes the matrix factorization implementation, section 4 presents the detection of leaders and the group-recommendation, and evaluation is detailed in section 5. Next, we discuss related work.

2. RELATED WORK

Although recommender systems have recently attracted a lot of attention from the scientific community, group recommendation has not been widely addressed, since most recommendation techniques are oriented to individual users and focus on maximizing the accuracy of their preference predictions. A. Jameson et al. [2] conducted an enlightening survey in 2007 presenting the most relevant works on the field of group recommendation, as well as the most common issues addressed by the authors of the surveyed group recommender systems. The main challenges faced when providing group recommendations are (1) capturing user preferences, (2) combining user preferences into a representation of group preferences, (3) defining criteria to assess the adequacy of recommendations, and (4) delivering recommendations. Group recommender systems can be compared according to how they deal with these challenges. In 2002, the Flytrap system was proposed by A. Crossen et al. [1], presenting a simple system designed to build a soundtrack that would please all users within a group in a target environment. In Flytrap system, user preferences were obtained by registering what they listen to on their private computers, in an implicit fashion. Recommendations were then computed by comparing songs within the system database to those listened to by the group members based on artist and genre. Songs whose artist or genre are known to please more users within the target group were then more eligible to be recommended automatically, without the user having any control over what's being recommended. The content-based nature of recommendations provided by Flytrap is a constant in most group recommender systems described in literature. A similar approach was taken in the system CATS (Collaborative Advisory Travel System) by McCarthy et al. [7]. CATS is a system designed to recommend travel packages to groups of users. It relies on a form of user feedback named *critiquing*, which consists in having the group users give their real-time opinion about some features associated with the recommended products in a *more of this / less of that* fashion. For example, when presented with a travel package recommendation a user can let the system know about his preference for a cheaper or shorter plan, without specifying price or duration values. This user feedback is recorded and linearly combined between all users within the group to be afterwards compared against the set of features that represent each travel package. The CATS systems can then recommend the travel packages that suit better the groups' critiques. Another example of group recommender systems is the system Bluemusic proposed by Mahato et al. [6]. In this system users are detected via bluetooth and the awareness of their presence has direct influence on a playlist which is being played on a public place. To be taken into account, a user must register his preferences beforehand. The concept introduced by the Bluemusic system is very simple but introduces an interesting alternative for incorporating transient awareness of user presences into a real-time playlist recommendation scenario.

In presented approaches either a content-based method or a simple collaborative approach merging the ratings of users was applied, which contrasts with our method that we tackles the problem in the latent factor space.

3. THE USER-MEDIA LATENT SPACE

In the context of recommender systems, matrix factorization is mainly performed through methods that approximate Singular Value Decomposition (SVD). SVD is a technique to decompose a matrix into the product $U\Sigma V$, where U contains the left singular vectors, Σ contains the singular values and V contains the right singular vectors of the original matrix. The application of SVD to recommender systems is motivated by the desire of decomposing the ratings matrix into a 2-matrices representation, as in eq. 2:

$$R = P \cdot Q^T \tag{2}$$

Here, matrix R is the ratings matrix where each r_{ui} value represents a rating given by user u to product i, expressed by a real value. In this modified version of SVD, we have $P = U \cdot \sqrt{\Sigma}$ and $Q = \sqrt{\Sigma} \cdot V$. Each vector (row) p_u of P represents a user u and each vector (row) q_i of Q represents a product i, as in eq. 1. The goal of using matrix factorization in recommendation problems is to enable the assessment of user preferences for products by calculating the dot product of their factor representations, as defined by eq. 3:

$$r_{u,i} = p_u \cdot q_i^T, \tag{3}$$

Here, $r_{u,i}$ is the preference of user u for product i, both represented as vectors as described in eq. 1. Based on the time-aware model proposed by Y. Koren [5], the minimization to be pursued is modified into:

$$[P,Q] = \arg\min_{p_u, q_i} \sum_{r_{ui}(t) \in R} (r_{ui}(t) - \mu - b_u(t) - b_i(t) - p_u \cdot q_i^T)^2 + \lambda \cdot (\|p_u\|^2 + \|q_i\|^2 + b_u(t)^2 + b_i(t)^2) \tag{4}$$

where μ is the mean rating average, $b_u(t)$ represents user biases and $b_i(t)$ represents product biases.

4. GROUP-BASED RECOMMENDATIONS

When there is more than one user to please, recommendations must be provided in a different way so that the whole group of people is satisfied. Our approach to group-based recommendation considers a small number of users and a set of leaders to smooth recommendations. The six main steps of the group recommendation process can be described next, see Figure 1. First, a predictive model based on SVD is obtained (step 1) and from it groups of users are discovered (step 2) applying k-means algorithms over the latent factor space created. Afterwards target users are detected within the context (step 3) and the system identifies the groups which these users belong (step 4). Finally, the preferences of the detected users and respective group leaders are combined (step 5) and the products that better match those preferences are recommended (step 6).

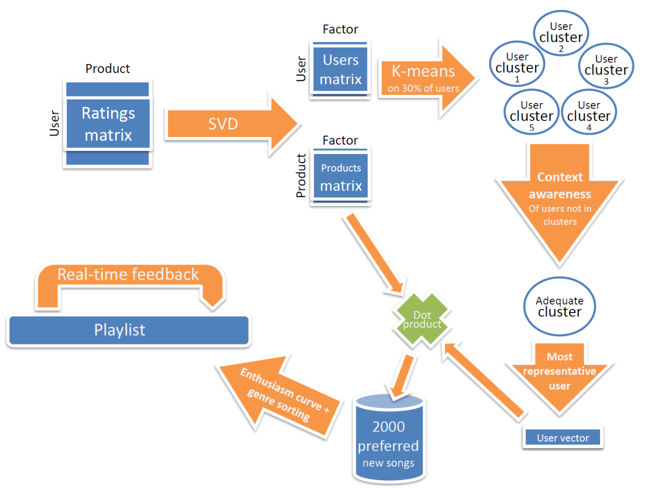

Figure 1: The proposed algorithm dataflow.

4.1 Discovering groups of users

The discovery of groups of user, which corresponds to step 2 of the algorithm, is performed after the matrix factorization stage, when user-factor and product-factor matrices are already computed. Group discovery was performed with three variants of the k-means (Lloyd's) algorithm. A well-known setback of this technique is that the k-means algorithm can get stuck at local minima, far from the optimal solution. For this reason it is common to consider heuristics based on local search, in which centroids are randomly swapped in and out of an existing solution. New solutions are accepted if they decrease the average distortion, and otherwise they are ignored. It is also possible to combine these two approaches (Lloyd's algorithm and local search), producing a type of hybrid solution. In this paper we used Mount's [3, 4] implementation of k-means. Besides standard k-means, we also applied two other variants of k-means, as listed below:

- **Swap:** A local search heuristic, which works by performing swaps between existing centroids and a set of candidate centroids.

- **Hybrid:** A more complex hybrid of Standard k-means and Swap, which performs some number of swaps followed by some number of iterations of algorithm.

The distance measure between centroids and data points, in this case users, used on this implementation was the Euclidean distance. This choice was motivated by the fact that during the matrix factorization process, the minimization earlier described in eq. 4 is obtained by recurrently measuring the squared euclidean distance to compute the latent space (i.e., the $(r_{ui} - p_u \cdot q_i^T)^2$ part in eq. **??**). In the end of each run, smaller clusters were eliminated and the users assigned to these were reallocated to the nearest cluster. The minimum number of users per cluster was set to 150.

4.2 Computing group-based recommendations

Computing group-based preferences comprises detecting the presence of users (step 3), relating them to the respective groups (step 4) and combining the preferences of those users with the preferences of their groups, represented by group leaders (step 4). If the system already has some knowledge regarding these detected users' preferences, such knowledge shall be used to produce a playlist.

An *early fusion* approach was taken, where the combination of target users latent-factor vectors was performed through a linear weighted combination, assigning more weight to more participative users, i.e., users that gave more ratings to products, as expressed by eq. 5.

$$g = \sum_i^n (\alpha_{u_i} \cdot u_i) \tag{5}$$

After detecting users and assessing to which groups these belong, the system identifies the group leaders. For each group, the group leader is the user that is closer to the respective cluster centroid, thus rendering this user as the most representative one. Group leaders' preferences are used to add diversity and smooth the group recommendations by combining these with detected users' preferences, extending eq. 5 into eq. 6:

$$g = \sum_i^n (\alpha_{u_i} \cdot u_i + \alpha_{l_i} \cdot l_i) \tag{6}$$

In eqs. 5 and 6, n is the number of target users and α_{u_i} and α_{u_i} are the weights assigned to the factor vectors of target user u_i and group leader l_i. As mentioned earlier, the weights assigned to each target user and group leader latent-factor vectors depend on the relation between the number of ratings given to products by these users and leaders and the total number of ratings given by all referred users and leaders $\alpha_u = \frac{nRatings_u}{totalGroupRatings}$.

Once obtained the latent-factor vector representing the users and leaders combined preferences, recommendations can be computed (step 6) by calculating the dot product between this vector and all product vectors contained in the product-factor matrix as expressed by eq. 7:

$$r_{g,i} = g \cdot q_i^T, \tag{7}$$

Now we can assess which products are more likely to satisfy this group. The products with higher predicted score can then be selected to build a playlist.

5. EVALUATION

The dataset used for this experiment is the one provided by Yahoo! to contestants of the KDD Cup 2011. From this dataset we used 3,867,473 ratings given by 20,000 users to 624,961 products to train the temp-SVD model and 80,000 ratings given by the same 20,000 users for validation during the learning process. All rating entries have a timestamp that was used to capture the temporal dynamics. The metric chosen to assess the accuracy of the time-SVD model was the root mean squared error (RMSE), which is the most commonly used accuracy measure for recommender systems.

5.1 Group discovery

Tests to evaluate the different variants of k-means assessed the division of users into clusters. Standard k-means, Swap and Hybrid variants were tested with 200 initial random centroids and 15,000 users, leaving out the remaining 5,000 users for further tests. Fig. 2 illustrates a comparison between the results obtained for the k-means variants Standard k-means, Swap and Hybrid.

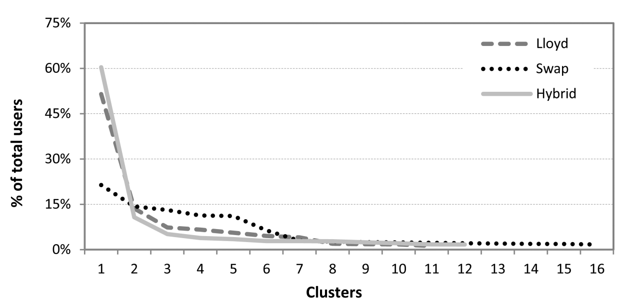

Figure 2: Comparison between Standard k-means, Swap and Hybrid variants of k-means clustering.

The Standard k-means algorithm broke de set of users down into 11 clusters. As we can see, its larger cluster contains more than 50% of all 15,000 users, which suggests that

	Single-user + leader			Group of users		
	k-Means	Swap	Hybrid	k-Means	Swap	Hybrid
Accuracy	76.5%	77.3%	73.6%	68.0%	77.4%	70.8%
Users per Clusters	n.a.	n.a.	n.a.	1404	1501	2783
Products per Clusters	n.a.	n.a.	n.a.	45	118	83
Common-ratings	n.a.	n.a.	n.a.	103	360	233

Table 1: Accuracy of group recommendation for different variants of k-means.

standard k-means cannot capture the details of the data structure in the latent space. Obtaining one very large cluster is a recurrent issue in clustering problems, and that is one of the reasons behind our decision of experimenting different variants. The Hybrid variant of k-means clustering yielded similar results to those obtained with Standard k-means algorithm, which possibly suggests a poor performance of the clustering process. However, the Swap variant produces the best cluster model, breaking the set of users down into 16 more balanced clusters, with its larger cluster containing only 21.4% of the total number of users. Thus, the underlying data structure is better captured by the Swap variant.

5.2 Group-based recommendation

The evaluation criteria used to assess the quality of playlists was the percentage of positive ratings given to songs (on the validation set only) by the users that belong to the target groups, within the set of songs contained in the recommended playlists. A rating equal to or higher than 3, on a scale from 0 to 5, is considered a positive rating. The statistics of clusters and users involved, as well as the ratios of rated songs within the recommended playlists songs will be disclosed along this evaluation process, in an attempt to provide a reasonable notion of the confidence levels associated with these experiments.

The chart illustrated by Fig. 3 establishes a comparison between the results of group recommendation oriented to single users and smoothed by taking into account their respective group leaders, and group recommendation oriented to several users from different groups. Moreover, the 3 variants of k-means were tested. Overall, the Swap variant of

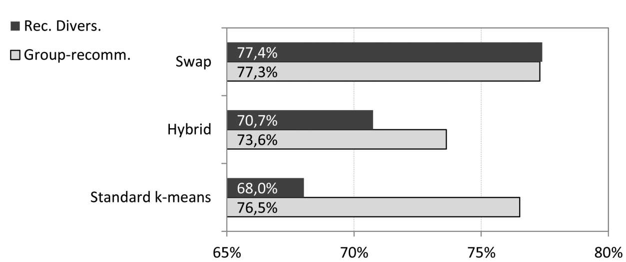

Figure 3: Comparison between single-group and multi-group recommendations using 3 different variants of k-means.

k-means obtained higher percentages of good recommendations. Additionally, for the Standard k-means and Hybrid variants, recommendations oriented to several users from different groups obtained higher percentages those oriented to single users We observe that the success of this system is dependent on whether the clustering algorithm breaks the set of users into meaningful groups or not. As we can observe on Fig. 2, which illustrates the results of k-means clustering with the Standard k-means variant, the largest cluster holds more than 50% of the total number of users, which may eventually indicate an inefficient clustering. If that's the case, such inefficient clustering would inevitably carry a negative impact to the resulting recommendations.

6. DISCUSSION

In this paper we addressed predictive models based on latent factor approaches, namely matrix factorization inspired by SVD, taking into account the rating biases and temporal fluctuations. We also addressed clustering by performing variants of the k-means algorithm over the latent space representation of users and the combination of users' and group leaders' preferences to obtain group-oriented recommendations. The use of latent space to solve group recommendation problems introduced in this paper was our most significant accomplishment, encouraging further research on the subject.

7. REFERENCES

[1] A. Crossen, J. Budzik, and K. J. Hammond. Flytrap: intelligent group music recommendation. In *International Conference on Intelligent User Interfaces*, IUI '02, pages 184–185, New York, NY, USA, 2002.

[2] A. Jameson and B. Smyth. The adaptive web. chapter Recommendation to groups, pages 596–627. Springer-Verlag, Berlin, Heidelberg, 2007.

[3] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24:881–892, July 2002.

[4] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. A local search approximation algorithm for k-means clustering. In *Annual Symposium on Computational Geometry*, SCG '02, pages 10–18, New York, NY, USA, 2002. ACM.

[5] Y. Koren. Collaborative filtering with temporal dynamics. *Commun. ACM*, 53:89–97, April 2010.

[6] H. Mahato, D. Kern, P. Holleis, and A. Schmidt. Implicit personalization of public environments using bluetooth. In *CHI '08 extended abstracts on Human factors in computing systems*, CHI EA '08, pages 3093–3098, New York, NY, USA, 2008. ACM.

[7] K. McCarthy, M. Salamó, L. Coyle, L. McGinty, B. Smyth, and P. Nixon. Cats: A synchronous approach to collaborative group recommendation. In *International Florida Artificial Intelligence Research Society Conference, Melbourne Beach, Florida, USA, May 11-13, 2006*, pages 86–91. AAAI Press, May 2006.