Impact of Content Novelty on the Accuracy of a Group Recommender System*

Ludovico Boratto and Salvatore Carta

Dipartimento di Matematica e Informatica,
Università di Cagliari, Via Ospedale 72 - 09124 Cagliari, Italy
{ludovico.boratto,salvatore}@unica.it

Abstract. A group recommender system is designed for contexts in which more than a person is involved in the recommendation process. There are types of content (like movies) for which it would be advisable to recommend an item only if it has not yet been consumed by most of the group. In fact, it would be trivial and not significant to recommended an item if a great part of the group has already expressed a preference for it. This paper studies the impact of content novelty on the accuracy of a group recommender system, by introducing a constraint on the percentage of a group for which the recommended content has to be novel. A comparative analysis in terms of different values of the percentage of the group and for groups of different sizes, was validated through statistical tests, in order to evaluate when the difference in the accuracy values is significant. Experimental results, deeply analyzed and discussed, show that the recommendation of novel content significantly affects the performances only for small groups and only when content has to be novel for the majority of it.

Keywords: Group Recommendation, Content Novelty, Clustering, Accuracy.

1 Introduction

Recommender systems aim to provide information items that are expected to interest a user [1]. *Group recommendation* is a type of recommendation designed for contexts in which more than a person is involved in the recommendation process [2, 3]. Group recommender systems suggest items to a group, by combining individual models that contain a user's preferences [4].

Group recommendation and detection of groups. A particular application scenario in which group recommendation is useful is when the number of recommendations that can be generated by a system is limited.

* This work is partially funded by Regione Sardegna under project SocialGlue, through PIA - Pacchetti Integrati di Agevolazione "Industria Artigianato e Servizi" (annualità 2010).

> *A company decides to print recommendation flyers that present suggested products. Even if the data to produce a flyer with individual recommendations for each customer is available, printing a different flyer for everyone would be technically too hard to accomplish and costs would be too high. A possible solution would be to set a number of different flyers to print, such that the printing process could be affordable in terms of costs and the recipients of the same flyer would be interested by its content.*

With respect to classic group recommendation, this type of systems adds the complexity of optimally defining groups, in order to respect the constraint on the number of recommendations that can be produced and maximize users' satisfaction. In the literature no system is able to automatically adapt to such constraints imposed by the system.

Recommendation of novel content. A group recommendation approach that recommends the same content previously evaluated by users would be useful for content that is always renewed and ever-changing, like news items or TV series episodes, since the user preferences for such types of content can be used to recommend items of the same type (e.g., news about the same topic or new episodes of the same series).

On the contrary, when a system produces group recommendations for types of content like movies, a new issue arises: the *novelty* of the recommended items. In fact, if an item was already evaluated by a great part of the group, the system should limit its recommendation, since users who already considered the item would be bored to reconsider it. From the user perspective, a group recommendation of an already evaluated item would be uninteresting and not significant, while from the system perspective it would be trivial to recommend items evaluated by a large part of the group. The recommendation of novel content is a key aspect that is being investigated both in the collaborative filtering [5] and the content-based [6] recommender systems literature.

Our contributions. This paper studies the impact of content novelty on the accuracy of a group recommended system. Recommending novel content creates a trade-off that involves an improvement in the satisfaction of the users and a loss in the accuracy of the predicted ratings. Since groups of different sizes are automatically detected by the system we used, this study allows a content provider to explore such a trade-off, by controlling the level of personalization of the recommended content. To the best of our knowledge, this is the first study of this type.

The scientific contributions of our study are the following:

- this is the first time that content novelty is studied in the group recommendation literature;
- novelty is evaluated for groups of different sizes and by considering different amounts of users for which a recommended item has to be novel. This allows to evaluate how the accuracy of the system evolves when the constraints change; that is, how content novelty affects the accuracy for groups of different sizes;

– a critical discussion of the obtained results is presented, in order to help in the design of a group recommender system that detects groups and allow a content provider to control the novelty of the recommended content.

The rest of the paper is organized in the following way: Section 2 presents related work in group recommendation; Section 3 contains a description of the system used in this study; Section 4 describes the experiments we conducted, outlines main results and presents a discussion of the results; Section 5 will draw conclusions and present future work.

2 Related Work

PolyLens [7], produces recommendations for groups of users who want to see a movie. A Collaborative Filtering approach is used to produce recommendations for each user of the group. The movies with the highest recommended ratings are considered and a "least misery" strategy is used, i.e., the recommended rating for a group is the lowest predicted rating for a movie, to ensure that every member is satisfied.

MusicFX [8] is a system that recommends music to the members of a fitness center. Since people in the room change continuously, the system gives the users that are working out in the fitness center the possibility to login. The music to play is selected considering the preferences of each user in a summation formula.

Flytrap [9] similarly selects music to be played in a public room. A 'virtual DJ' agent is used by the system to automatically decide the song to play. The agent analyzes the MP3 files played by a user in her/his computer and considers the information available about the music (like similar genres, artists, etc.). The song to play is selected through a voting system, in which an agent represents each user in the room and rates the candidate tracks.

In-Vehicle Multimedia Recommender [10] is a system that aims at selecting multimedia items for a group of people traveling together. The system aggregates the profiles of the passengers and merges them, by using a notion of distance between the profiles. A content-based system is used to compare multimedia items and group preferences.

FIT (Family Interactive TV System) [11] is a TV program recommender system. The only input required by the system is a stereotype user representation (i.e., a class of viewers that would suit the user, like *women*, *businessmen*, *students*, etc.), along with the user preferred watching time. When someone starts watching TV, the system looks at the probability of each family member to watch TV in that time slot and predicts who there might be watching TV. Programs are recommended through an algorithm that combines such probabilities and the user preferences.

TV4M [12] recommends TV programs for multiple viewers. The system identifies who is watching TV, by providing a login feature. In order to build a group profile that satisfies most of its members, all the current viewers profiles are merged, by doing a total distance minimization of the features available (e.g., genre, actor, etc.). According to the built profile, programs are recommended to the group.

In [13] a group recommender system called *CATS (Collaborative Advisory Travel System)* is presented. Its aim is to help a group of friends plan and arrange ski holidays. To achieve the objective, users are positioned around a device called "DiamondTouch table-top" and the interactions between them (since they physically share the device) help the development of the recommendations.

Pocket RestaurantFinder [14] is a system that suggests restaurants to groups of people who want to dine together. Each user fills a profile with preferences about restaurants, like the price range or the type of cuisine they like (or do not like). Once the group composition is known, the system estimates individual preference for each restaurant and averages those values to build a group preference and produce a list of recommendations.

Travel Decision Forum [15] is a system that helps groups of people plan a vacation. Since the system aims at finding an agreement between the members of a group, asynchronous communication is possible and, through a web interface, a member can view (and also copy) other members preferences. Recommendations are made by using the median of the individual preferences.

In [16], Chen and Pu present *CoFeel*, an interface that allows to express through colors the emotions given by a song chosen by the *GroupFun* music group recommender system. The interface allows users to give a feedback about how much they liked the song and the system considers the preferences expressed through the emotions, in order to generate a playlist for a group.

In [17], Jung develops an approach to identify long tail users, i.e., users who can be considered as experts on a certain attribute. So, the ratings given by the long tail user groups are used, in order to provide a relevant recommendation to the non-expert user groups, which are called short head groups.

Our approach differs from the ones in the literature, since none of the existing group recommendation approaches works with automatically detected groups and is able to adapt to constraints imposed by the context in which a system operates. Moreover, no work ever conducted a study to evaluate the impact of novelty on the accuracy of a group recommender system.

3 Group Recommendation With Automatic Detection of Groups

This section describes the group recommender system used for this study, named *Predict&Cluster*, which automatically detects groups by clustering users.

The tasks performed by the systems are the following:

1. *Predictions of the missing ratings for individual users.* Predictions are built for each user with a User-Based Collaborative Filtering Approach.
2. *Detection of the groups.* Considering both the individual preferences expressed by each user and the predicted ratings, groups of similar users are detected with the k-means clustering algorithm.

3. *Group modeling.* Once groups have been detected, a group model is built for each group, by using the *Additive Utilitarian* modeling strategy.

All the tasks are now be described in detail.

3.1 Predictions of the missing ratings for individual users

The missing ratings for the items not evaluated by each user are predicted with a classic User-Based Nearest Neighbor Collaborative Filtering algorithm, presented in [18]. This has been proved to be the most accurate way to predict ratings in this scenario [19].

The algorithm predicts a rating p_{ui} for each item i that was not evaluated by a user u, by considering the rating r_{ni} of each similar user n for the item i. A user n similar to u is called a *neighbor* of u. Equation (1) gives the formula used to predict the ratings:

$$p_{ui} = \overline{r}_u + \frac{\sum_{n \subset neighbors(u)} userSim(u,n) \cdot (r_{ni} - \overline{r}_n)}{\sum_{n \subset neighbors(u)} userSim(u,n)} \tag{1}$$

Values $\overline{r}_u$ and $\overline{r}_n$ indicate the mean of the ratings given by user u and user n. Similarity $userSim()$ between two users is calculated using the Pearson's correlation, which compares the ratings of the items rated by both the target user and the neighbor. Pearson's correlation among a user u and a neighbor n is given in Equation (2) (I_{un} is the set of items rated by both u and n).

$$userSim(u,n) = \frac{\sum_{i \subset I_{un}} (r_{ui} - \overline{r}_u)(r_{ni} - \overline{r}_n)}{\sqrt{\sum_{i \subset I_{un}} (r_{ui} - \overline{r}_u)^2} \sqrt{\sum_{i \subset I_{un}} (r_{ni} - \overline{r}_n)^2}} \tag{2}$$

The metric ranges between 1.0 and -1.0. Negative values do not increase the prediction accuracy [20], so they are discarded by the task.

3.2 Detection of the groups

In order to respect the constraint imposed by the context, the set of users has to be partitioned into a number of groups equal to the number of recommendations that can be produced. Since in our application scenario groups do not exist, unsupervised classification (*clustering*) is necessary.

In [21] it was highlighted that the sparsity of the rating matrix strongly affects the performances of a group recommender system and, in particular, of the clustering task. Therefore, a group recommender system that clusters users based on the individual ratings has to include personal predictions in the clustering input; this allows to avoid sparsity, overcome the curse of dimensionality, and increase the system accuracy.

In [22], authors highlight that the k-means clustering algorithm is by far the most used clustering algorithm in recommender systems. Moreover, in previous studies we analyzed [23] and compared [24] a different option to group the users, by using the Louvain community detection algorithm, which produces a hierarchical partitioning of the users; however, results showed that k-means is more accurate in this context.

This task clusters users with the k-means clustering algorithm, based on the individual preferences explicitly expressed by them and on the individual predictions built by the previously presented task. The output produced is a partitioning of the users into groups (clusters), such that users with similar models (i.e., similar ratings for the same items) are in the same group and can receive the same recommendations.

3.3 Group modeling

In order to create a model that represents the preferences of a group, the *Additive Utilitarian* group modeling strategy [4] is adopted. The strategy sums the individual ratings for each item and produces a list of group ratings (the higher the sum is, the earlier the item appears in the list). The ranked list of items is exactly the same that would be produced when averaging the individual ratings, so this strategy is also called 'Average strategy'. An example of how the strategy works is given in Table 1. The example considers three users (u_1, u_2 and u_3) that rate eight items (i_1, ..., i_8) with a rating from 1 to 10.

In a recent work [25] we showed that this metric allows to obtain the most accurate performances for a group recommender system that detects groups. Indeed, results show that:

- since the considered scenario deals with a limited number of recommendations, the system works with large groups. Therefore, an average, which is a single value that is meant to typify a set of different values, is the best way to put together the ratings in this context;
- for groups created with the k-means clustering algorithm, creating a group model with an average of the individual values for each item is like re-creating the centroid of the cluster, i.e., a super-user that connects every user of the group.

In order to have the same scale of ratings both in the group models and in the individual user models, the produced group models contain the average of the individual predictions, instead of the sum.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
u_1	8	10	7	10	9	8	10	6
u_2	7	10	6	9	8	10	9	4
u_3	5	1	8	6	9	10	3	5
Group	20	21	21	25	26	28	22	15

Table 1. Example of the *Additive Utilitarian* strategy

4 Experimental Framework

This section presents the framework built for the experiments.

4.1 Experimental Setup

To conduct the experiments, we adopted the MovieLens-1M dataset.

The number of neighbors used by the first task to predict the ratings is 100 (see [19] for the details of the experiments that allowed to set the value).

The clusterings with k-means were created using a testbed program called KMlocal [26], which contains a variant of the k-means algorithm, called *EZ Hybrid*, that was chosen because it returned a lowest average distortion.

The RMSE values obtained by the system considering increasing percentages of the group for which content had to be novel have been compared, by considering different numbers of groups to detect (that correspond to the number of recommendations that can be produced by the system). The choice to measure the performances for different numbers of groups has been made to show how the accuracy of the systems changes as the constraint changes. In order to evaluate the quality of the predicted ratings for different numbers of groups, in each experiment four different clusterings of the users into 20, 50, 200 and 500 groups were created. Moreover, we compared the results obtained with the previously mentioned four clusterings with the results obtained considering a single group with all the users (i.e., we tested the system in a scenario in which just one set of recommendations can be produced, so predictions for an item are calculated considering the preferences of all the users), and the results obtained by the system that calculates individual predictions for each user (i.e., we simulated the case where there is no constraint, in order to compare the performances of the algorithms when they work with groups).

RMSE was chosen to compare the systems because it is widely used, allows to evaluate results through a single number and emphasizes large errors.

In order to evaluate if two RMSE values returned by two experiments are significantly different, independent-samples two-tailed Student's t-tests have been conducted. In order to make the tests, a 5-fold cross-validation was preformed.

In each experiment, we evaluated the system performances considering different values of a *novelty* parameter, which expresses the minimum percentage of users in a group that did not previously rate an item, in order for it to be recommended. For example, if *novelty* was set to 50% and an item was rated by 60% of the group, the predicted rating for that item would be discarded, since it would be novel just for 40% of the group.

4.2 Dataset and Data Preprocessing

The MovieLens-1M[1] dataset contains 1 million ratings, given by 6040 users for 3900 movies. Our framework uses only the file `ratings.dat`, which contains

[1] http://www.grouplens.org/

the user ratings. The file was preprocessed by mapping the feature *UserID* into a new set of IDs between 0 and 6039 to facilitate the computation with data structures. In order to conduct the cross-validation, the dataset was split into five subsets with a random sampling technique (each subset contains 20% of the ratings).

4.3 Metrics

The quality of the predicted ratings was measured through the Root Mean Squared Error (RMSE). The metric compares each rating r_{ui}, expressed by a user u for an item i in the test set, with the rating p_{gi}, predicted for the item i for the group g in which user u is. The formula is shown below:

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n}(r_{ui} - p_{gi})^2}{n}}$$

where n is the number of ratings available in the test set. In order to compare if two RMSE values returned by two experiments are significantly different, independent-samples two-tailed Student's t-tests have been conducted. These tests allow to reject the null hypothesis that two values are statistically the same. So, a two-tailed test will evaluate if an RMSE value is significantly greater or significantly smaller than another RMSE value. Since each experiment was conducted five times, the means M_i and M_j of the RMSE values obtained by two systems i and j are used to compare the systems and calculate a value t:

$$t = \frac{M_i - M_j}{s_{M_i - M_j}}$$

where

$$s_{M_i - M_j} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$$

s^2 is the variance of the two samples, n_1 and n_2 indicate the number of values considered to build M_1 and M_2 (in our case both are equal to 5, since experiments were repeated five times). In order to determine the $t - value$ that indicates the result of the test, the degrees of freedom have to be determined:

$$\text{d.f.} = \frac{(s_1^2/n_1 + s_2^2/n_2)^2}{(s_1^2/n_1)^2/(n_1 - 1) + (s_2^2/n_2)^2/(n_2 - 1)}$$

Given t and $d.f.$, the $t - value$ (i.e., the results of the test), can be obtained in a standard table of significance as: $t(d.f.) = t - value$. The $t - value$ derives the probability p that there is no difference between the two means. Along with the result of a t-test, the standard deviation SD of the mean is presented.

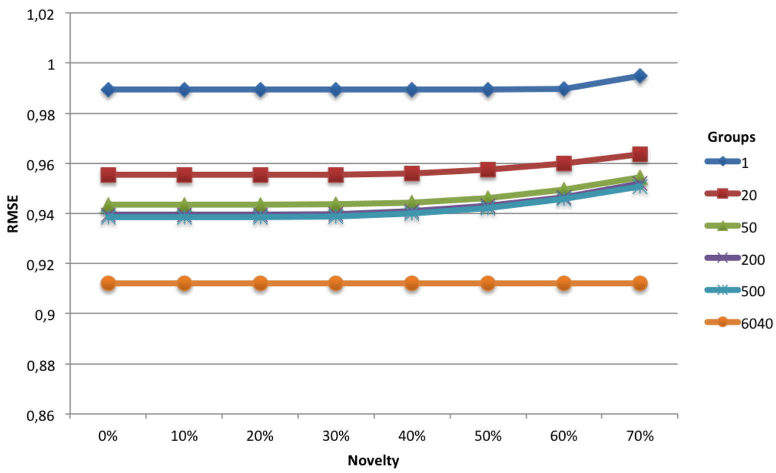

Fig. 1. Performances for different values of novelty

4.4 Experimental Results

Figure 1 shows the RMSE for different values of the *novelty* parameter and each set of groups.

Since the results between the different settings are very close, we performed an independent-samples two-tailed Student's t-test for each point in the figure, in order to compare if the difference in the results between no *novelty* ($novelty =$ 0%) and a specific *novelty* value was significant. Because of the amount of tests conducted, we summarized the results in Table 2.

Each cell of the table contains the result of a test, which is the probability p that the difference in the RMSE values obtained with two different *novelty* values for a partitioning is due to chance. For example, $p(n = 0\%\ n = 10\%, g = 1) = 1.00$, means that for one group ($g = 1$), the difference in the RMSE values obtained with $novelty = 0\%$ and $novelty = 10\%$ is not significant (in fact the probability p that the two values are the same is 1).

4.5 Discussion

Results show that up to 40% there is no significant worsening of the performances ($p \geq 0.5$). These really good results mean that if we avoid recommending all the items that have already been evaluated by at least 40% of the users in a group, the system is still able to produce accurate recommendations to the group. In other words, our system is still able to keep the same accuracy, while increasing user satisfaction, by filtering the recommended content.

Performances start worsening when $novelty > 40\%$ and for a high number of groups ($g > 20$ in Table 2). So, when the number of groups is high and groups

	g=1	g=20	g=50	g=200	g=500
n=0% n=10%	1.00	1.00	1.00	1.00	0.99
n=0% n=20%	1.00	1.00	1.00	1.00	0.99
n=0% n=30%	1.00	1.00	0.95	0.75	0.64
n=0% n=40%	1.00	0.85	0.49	0.11	0.05
n=0% n=50%	1.00	0.44	0.06	0.00	0.00
n=0% n=60%	0.66	0.10	0.00	0.00	0.00
n=0% n=70%	0.00	0.01	0.00	0.00	0.00

Table 2. Results of the the independent-samples two-tailed Student's t-tests

get smaller, worsening of the results is faster, i.e., small groups are more affected by content novelty and it is harder to recommend new items to a small group.

So, when designing a group recommender system that automatically detects groups, the number of recommendations that can be generated and the consequent number of groups detected by the system is the key aspect that guides the amount of content novelty that can be provided to users. More specifically, novel content can be provided to at least 40% the group, no matter how many groups are detected by the system. If the system deals with a large amount of groups, the system is not able to recommend novel content without affecting the results.

5 Conclusions and Future Work

In this paper we presented a study related to the recommendation of novel content in a group recommender system that detects groups, in order to respect a constraint on the number of recommendations that can be produced.

We conducted an analysis in order to study how the performances of the system are affected when a constraint on the percentage of a group for which content has to be novel is introduced.

Experimental results, validated through statistical tests, show that our system is able to recommend novel content to nearly half of a group without affecting its accuracy. This means that user satisfaction can be increased for a good part of a group with no cost in terms of performances.

The fact that the same percentage of the *novelty* parameter leads to significantly different performances according to the structure of the group, opens a new research scenario, related to finding the properties of a group that characterize the performances of the system. This analysis is left as future work.

References

1. Ricci, F., Rokach, L., Shapira, B.: Introduction to recommender systems handbook. In: Recommender Systems Handbook. Springer, Berlin (2011) 1–35
2. Boratto, L., Carta, S.: State-of-the-art in group recommendation and new approaches for automatic identification of groups. In: Information Retrieval and Mining in Distributed Environments. Springer Berlin Heidelberg (2011) 1–20

3. Jameson, A., Smyth, B.: Recommendation to groups. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web, Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science., Springer (2007) 596–627
4. Masthoff, J.: Group recommender systems: Combining individual models. In: Recommender Systems Handbook. Springer (2011) 677–702
5. Ziegler, C.N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In Ellis, A., Hagino, T., eds.: Proceedings of the 14th international conference on World Wide Web, WWW 2005, Chiba, Japan, May 10-14, 2005, ACM (2005) 22–32
6. Lops, P., de Gemmis, M., Semeraro, G.: Content-based recommender systems: State of the art and trends. In Ricci, F., Rokach, L., Shapira, B., Kantor, P.B., eds.: Recommender Systems Handbook. Springer (2011) 73–105
7. O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J.: Polylens: A recommender system for groups of user. In: Proceedings of the Seventh European Conference on Computer Supported Cooperative Work, Kluwer (2001) 199–218
8. McCarthy, J.F., Anagnost, T.D.: Musicfx: An arbiter of group preferences for computer supported collaborative workouts. In Poltrock, S.E., Grudin, J., eds.: CSCW '98, Proceedings of the ACM 1998 Conference on Computer Supported Cooperative Work, Seattle, WA, USA, November 14-18, 1998, ACM (1998) 363–372
9. Crossen, A., Budzik, J., Hammond, K.J.: Flytrap: intelligent group music recommendation. In: Proceedings of the 7th international conference on Intelligent user interfaces. IUI '02, New York, NY, USA, ACM (2002) 184–185
10. Zhiwen, Y., Xingshe, Z., Daqing, Z.: An adaptive in-vehicle multimedia recommender for group users. In: Proceedings of the 61st Semiannual Vehicular Technology Conference. Volume 5. (2005) 2800–2804
11. Goren-Bar, D., Glinansky, O.: Fit-recommend ing tv programs to family members. Computers & Graphics **28**(2) (2004) 149–156
12. Yu, Z., Zhou, X., Hao, Y., Gu, J.: Tv program recommendation for multiple viewers based on user profile merging. User Modeling and User-Adapted Interaction **16**(1) (March 2006) 63–82
13. McCarthy, K., Salamó, M., Coyle, L., McGinty, L., Smyth, B., Nixon, P.: Cats: A synchronous approach to collaborative group recommendation. In Sutcliffe, G., Goebel, R., eds.: Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference, Melbourne Beach, Florida, USA, May 11-13, 2006, AAAI Press (2006) 86–91
14. McCarthy, J.: Pocket RestaurantFinder: A situated recommender system for groups. In: Workshop on Mobile Ad-Hoc Communication at the 2002 ACM Conference on Human Factors in Computer Systems. (2002)
15. Jameson, A.: More than the sum of its members: challenges for group recommender systems. In: Proceedings of the working conference on Advanced visual interfaces, ACM Press (2004) 48–54
16. Chen, Y., Pu, P.: Cofeel: Using emotions to enhance social interaction in group recommender systems. In: Alpine Rendez-Vous (ARV) 2013 Workshop on Tools and Technology for Emotion-Awareness in Computer Mediated Collaboration and Learning. (2013)
17. Jung, J.J.: Attribute selection-based recommendation framework for short-head user group: An empirical study by movielens and imdb. Expert Systems with Applications **39**(4) (March 2012) 4049–4054

18. Schafer, J.B., Frankowski, D., Herlocker, J.L., Sen, S.: Collaborative filtering recommender systems. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web, Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science., Springer (2007) 291–324
19. Boratto, L., Carta, S.: Exploring the ratings prediction task in a group recommender system that automatically detects groups. In: IMMM 2013, The Third International Conference on Advances in Information Mining and Management. (2013) 36–43
20. Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR '99, New York, NY, USA, ACM (1999) 230–237
21. Boratto, L., Carta, S.: Using collaborative filtering to overcome the curse of dimensionality when clustering users in a group recommender system. In: Proceedings of 16th International Conference on Enterprise Information Systems (ICEIS). (2014) 564–572
22. Amatriain, X., Jaimes, A., Oliver, N., Pujol, J.M.: Data mining methods for recommender systems. In Ricci, F., Rokach, L., Shapira, B., Kantor, P.B., eds.: Recommender Systems Handbook. Springer, Boston, MA (2011) 39–71
23. Boratto, L., Carta, S., Chessa, A., Agelli, M., Clemente, M.L.: Group recommendation with automatic identification of users communities. In: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 03. WI-IAT '09, Washington, DC, USA, IEEE Computer Society (2009) 547–550
24. Boratto, L., Carta, S., Satta, M.: Groups identification and individual recommendations in group recommendation algorithms. In Picault, J., Kostadinov, D., Castells, P., Jaimes, A., eds.: Practical Use of Recommender Systems, Algorithms and Technologies 2010. Volume 676 of CEUR Workshop Proceedings. (November 2010)
25. Boratto, L., Carta, S.: Modeling the preferences of a group of users detected by clustering: A group recommendation case-study. In: Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14). WIMS '14, New York, NY, USA, ACM (2014) 16:1–16:7
26. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: An efficient k-means clustering algorithm: Analysis and implementation. IEEE Trans. Pattern Anal. Mach. Intell. **24**(7) (2002) 881–892