Design Criteria to Model Groups in Big Data Scenarios: Algorithms and Best Practices*

Ludovico Boratto, Gianni Fenu, and Pier Luigi Pau

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

Abstract. There are different types of information systems, such as those that perform *group recommendations* and *market segmentations*, which operate with groups of users. In order to combine the individual preferences and properly address suggestions to users, *group modeling* strategies are employed. Nowadays, data is characterized by large amounts in terms of volume, speed, and variety (the so-called *big data* issue). In this paper, we are going to tackle the problem of modeling group preferences in big data scenarios. This study will present the existing strategies, and we are going to present criteria to design the algorithms that implement them when big amounts of data have to be combined. Moreover, a set of best practices discusses under which conditions the presented strategies can be adopted in big data scenarios.

Keywords: Group Modeling, Big Data, Algorithms, Design.

1 Introduction

Combining the preferences of individual users is a central problem for the information systems that operate with groups. The most challenging and widely studied, both by the industry and the academia, are the *group recommender* [1, 2] and *market segmentation* [3, 4] systems, which aggregate information about large groups of users and tens of items in order to filter the data and produce suggestions for the users in terms of items or ads. Therefore, nowadays these systems have to deal with big data and to be able to filter large amounts of information.

The task of aggregating the individual preferences into a single model is known as *group modeling*, and several strategies have been studied in the literature [5]. It is known that no strategy is better than another and that the

* This work is partially funded by Regione Sardegna under project SocialGlue, through PIA - Pacchetti Integrati di Agevolazione "Industria Artigianato e Servizi" (annualità 2010), and by MIUR PRIN 2010-11 under project "Security Horizons". Pier Luigi Pau gratefully acknowledges Sardinia Regional Government for the financial support of his PhD scholarship (P.O.R. Sardegna F.S.E. Operational Programme of the Autonomous Region of Sardinia, European Social Fund 2007-2013 - Axis IV Human Resources, Objective l.3, Line of Activity l.3.1.).

group modeling strategy adopted by an information system should be chosen after a deep analysis of the application domain in which the groups have to be modeled [6].

In this paper, we tackle the novel problem of *studying criteria for applicable and efficient design of group modeling algorithms in big data scenarios*. More specifically, we are going to answer the following research question: *which group modeling strategies can actually be employed in real-world contexts characterized by big data?* In order to answer this question, we first present the existing group modeling strategies (Section 2), then we propose design guidelines to efficiently implement these strategies in big data scenarios, and discuss with a set of best practices which strategies are applicable in real-world big data contexts (Section 3). Our aim is to guide future research in this area towards the development of approaches that are efficient and effective at the same time. This study is concluded with a summary of the proposed criteria and with perspectives for future work in this research area (Section 4).

2 Background and Related Work

Group modeling [5] is the process adopted to combine multiple user models into a single model. In this section, we are going to present the modeling strategies that have been employed in the literature. In order to facilitate their understanding, an example of the results produced by the strategies is given as a reference, then we present each of them.

2.1 Group Modeling: Working Examples.

Here, we present an example of how each group modeling strategy operates. We consider three users (denoted as u_1, u_2, and u_3), who rate ten items (i_1, ..., i_{10}) with a rating from 1 to 10. Table 1 reports the output of the strategies that combine individual ratings, while tables 2, 3, and 4, show how the *Borda Count*, *Copeland Rule*, and *Plurality Voting* strategies respectively work (these tables are based on the ratings in Table 1).

2.2 Additive Utilitarian [AU]

The individual ratings for each item are summed and a list of items ranked by sum is created. The list produced by each strategy is the same that would be generated when averaging the individual ratings, so it is also called 'Average strategy'. An example of how the strategy works is given in Table 1 (*AU* line).

The strategy has proven to be effective in different contexts [7], like the combination of preferences on different types of features (e.g., location, cost, cuisine) when recommending restaurants to a group [8].

2.3 Multiplicative Utilitarian [MU]

The ratings given by the users for each item are multiplied and a ranked list of items is produced. An example of how the strategy works is given in Table 1 (MU line).

This strategy was employed in the music recommendation domain by [9].

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	8	10	7	10	9	8	10	6	3	6
u_2	7	10	6	9	8	10	9	4	4	7
u_3	5	1	8	6	9	10	3	5	7	10
$-AU$	20	21	21	25	26	28	22	15	14	23
$-MU$	280	100	336	540	648	800	270	120	84	420
$-AV$	2	2	3	3	3	3	2	1	1	3
$-LM$	5	1	6	6	8	8	3	4	3	6
$-MP$	8	10	8	10	9	10	10	6	7	10
$-AWM$	20	-	21	25	26	28	-	15	-	23
$-MRP$	8	10	7	10	9	8	10	6	3	6

Table 1. Output of the strategies that combine the original ratings

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	4.5	8	3	8	6	4.5	8	1.5	0	1.5
u_2	3.5	7.5	2	6.5	5	7.5	6.5	0.5	0.5	3.5
u_3	2.5	0	5	3	6	7.5	1	2.5	4	7.5
$-BC$	10.5	15.5	10	17	17	19.5	15.5	4.5	4.5	12.5

Table 2. Example of how the *Borda Count* strategy works, based on the ratings in Table 1

2.4 Borda Count [BC]

The strategy assigns to an item a number of points, according to the position in the list of each user. The least favorite one gets 0 points and a point is added each time the next item in the list is considered. If a user gave the same rating to more than one item, the points are distributed. Considering the example in Table 2, items i_8 and i_9 were rated by user u_2 with the lowest rating and share the lowest positions with 0 and 1 points, by getting (0+1)/2=0.5 points. A group preference is obtained by adding the individual points of an item.

This strategy was implemented in [10].

2.5 Copeland Rule [CR]

It is a form of majority voting that sorts the items according to their *Copeland index*, which is calculated as the number of times in which an alternative beats

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
i_1	0	+	-	+	+	+	+	-	-	0
i_2	-	0	-	0	-	0	0	-	-	-
i_3	+	+	0	+	+	+	+	-	-	+
i_4	-	0	-	0	-	+	-	-	-	-
i_5	-	+	-	+	0	+	+	-	-	-
i_6	-	0	-	-	-	0	-	-	-	-
i_7	-	0	-	+	-	+	0	-	-	-
i_8	+	+	+	+	+	+	+	0	0	+
i_9	+	+	+	+	+	+	+	0	0	+
i_{10}	0	+	+	+	+	+	+	-	-	0
Index	-2	+6	-3	+6	+1	+8	+4	-8	-8	-2

Table 3. Example of how the *Copeland Rule* strategy works, based on the ratings in Table 1

	1	2	3	4	5	6
u_1	i_2, i_4, i_7	i_4, i_7	i_5	i_1	i_3	i_8
u_2	i_2, i_6	i_4, i_7	i_5	i_1, i_{10}	i_3	i_8, i_9
u_3	i_6, i_{10}	i_{10}	i_{10}	i_{10}	i_3	i_9
Group	i_2, i_6	i_4, i_7	i_5	i_1, i_{10}	i_3	i_8, i_9

Table 4. Example of how the *Plurality Voting* strategy works, based on the ratings in Table 1

the others, minus the number of times it loses against the other alternatives. In the example in Table 3, item i_2 beats item i_1, since it received a higher rating by both users u_1 and u_2.

The approach proposed in [11] proved that a form of majority voting is the most successful in a *requirements negotiation* context.

2.6 Plurality Voting [PV]

Each user votes for her/his favorite option. The one that receives the highest number of votes wins. If more than one alternative needs to be selected, the options that received the highest number of votes are selected. An example of how the strategy works is given in Table 4.

This strategy was implemented and tested by [12, 13] in the TV domain.

2.7 Approval Voting [AV]

Each user votes for as many items as she/he wants, and a point is assigned to all the ones a user likes. To show how the strategy works, in the example in Table 1 (*AV* line) we suppose that each user votes for all the items with a rating above a threshold (for example, 5). A group preference is obtained by adding the individual points of an item.

To choose the pages to recommend to a group, *Let's Browse* [14] evaluates if the page currently considered by the system matches with the user profile above a certain threshold and recommends the one with the highest score. This strategy also proved to be successful in contexts in which the similarity between the users in a group is high [15].

2.8 Least Misery [LM]

The group rating produced for an item is the lowest rating expressed for that item by any of the users in the group. This strategy usually models small groups, to make sure that every member is satisfied. A drawback is that if the majority of the group really likes something, but one person does not, the item will not to be recommended to the group. This is what happens in Table 1 for the items i_2 and i_7. An example of how the strategy works is given in Table 1 (LM line).

This strategy is used by [16], to recommend movies to small groups.

2.9 Most Pleasure [MP]

The rating assigned to an item for a group is the highest one expressed for that item by a member of a group. An example of how the strategy works is given in Table 1 (MP line).

This strategy is used by [17] in a system that faces the cold start problem.

2.10 Average Without Misery [AWM]

The rating assigned to an item for a group is the average of the ratings given by each user. All the items that received a rating under a certain threshold by a user are not included in the group model (in the example in Table 1 - AWM line, the threshold rating is 4).

In order to model a group to decide the music to play in a gym, in [18] the individual ratings are summed, discarding the ones under a minimum degree.

2.11 Fairness [F]

This strategy is based on the idea that users can be recommended something they do not like, as long as they also get recommended something they like. This is done by allowing each user to choose her/his favorite item. If two items have the same rating, the choice is based on the other users' preferences. This is done until everyone made a choice. Next, everyone chooses a second item, starting from the person who chose last the first time.

If in the example in Table 1, we suppose that user u_1 chose first, she/he would consider i_2, i_4, and i_7, and would choose i_4, because it has the highest average considering the other users' ratings. Next, u_2 would choose between i_2 and i_6 and would select i_6 for the same reason. Then, u_3 would choose item i_{10}. Since everyone chose an item, it would be u_3's turn again and i_5 would be chosen. User u_2 would choose i_2, which has the highest rating along with i_6 (which was already chosen). Then, u_1 would choose i_7, which is the one with the highest rating and was not chosen yet. The final sequence of items that models the group would be: i_4, i_6, i_{10}, i_5, i_2, i_7, i_1, i_3, i_9, i_8.

This strategy is adopted by [9] in the music recommendation context.

2.12 Most Respected Person (Dictatorship) [MRP]

This strategy selects the items according to the preferences of the most respected person, using the preferences of the others just in case more than one item received the same evaluation. The idea is that there are scenarios is which a group is guided/dominated by a person. In the example in Table 1, u_1 is the most respected person.

This strategy is adopted to select tourist attractions advantaging the users with particular needs [19], or when experts are recognized in a group [20]. Moreover, there are studies that highlight that when people interact, a user or a small portion of the group influences the choices of the others [21].

3 Criteria for Applicable Design of Group Modeling Algorithms in Big Data Scenarios

This section presents design criteria to implement the strategies presented in the previous section in scenarios characterized by big data. In Section 3.1, we are going to present design criteria from an algorithmic point of view, while in Section 3.2 we are going to study the nature of each strategy to evaluate their applicability in real-world scenarios characterized by big data.

3.1 Algorithms Design

Each strategy is fairly trivial to implement in an efficient way, by adopting data structures that can be quickly accessed. A possible way to implement a group model might be a *hash table* that stores the item ids as keys and the group rating as values. Each time a new individual rating arrives, the hash table can be efficiently updated with average complexity $O(1)$.

In order to suggest the items to the users, the items in the group model have to be sorted by group rating. An efficient sorting algorithm for big data, known as *two-way replacement selection*, has been proposed in [22]. The algorithm presents a variant of the Merge Sort algorithm, specifically designed for big data scenarios, and it currently represents the state of the art.

3.2 Applicability in Big Data Scenarios: Best Practices

Here, we present a set of best practices related to the applicability of the previously presented group modeling strategies in big data scenarios. Most of these best practices are derived from a case-study conducted in the group recommendation domain and presented in [23], and from considerations on the aspects that characterize big data scenarios.

The main argument against the deployment of a strategy in a big data scenario will be represented by a high computational cost of performing inserts and updates of ratings in large sets of data. More precisely, it is assumed that group ratings, resulting from the application of a specific strategy on a set of

user ratings, are stored for later use in order to save computation power, and that a recalculation of group ratings is required following the insertion or update of user ratings. This evaluation takes into account the computational cost of updating group ratings. Furthermore, for the sake of completeness, it will also be noted when a strategy is simply inadequate for modeling large groups of users, regardless of any difficulties in handling large amounts of data.

The *Additive Utilitarian* strategy can be easily employed in big data scenarios, as the only operation required to update the group model is to add the rating expressed by a user for an item to the existing group rating for that item.

Likewise, the *Multiplicative Utilitarian* would be very simple to implement. However, it would not be advisable to employ it in presence of large groups, as an overflow would most certainly occur when the original user ratings are multiplied[1]. Moreover, given the large amounts of operations that the strategy would perform for such a group, rating normalization to avoid the problem would lead to a loss in precision and to a drop in the accuracy of the system.

Borda Count, *Copeland Rule*, *Plurality Voting*, and *Fairness* require to update the individual model of each user each time she/he assigns a new rating. After the individual model is updated, the group model can be updated accordingly. Therefore, these strategies would not be efficiently applicable in big data scenarios.

The *Average Without Misery*, *Approval Voting*, *Least Misery*, and *Most Pleasure* strategies, can apparently be easily and efficiently adapted in big data scenarios, as they only require to calculate an average, the maximum, or the minimum of the individual ratings given to each item. However, they should not be adopted in big data due to their nature. Indeed, *Average Without Misery* discards a group rating if at least a user has given to an item a rating lower than the considered threshold. Therefore, even with a small threshold value, like 2, the vast majority of the items would not be modeled by the strategy in a context in which groups are large (the larger is the group, the higher is the number of times an item is rated, and higher is the probability that at least one user did not like the item). Considering the *Approval Voting* strategy with high threshold values (for example, 5), too many ratings would be discarded by the model because only the items with a high rating (i.e., with a rating above 5), would be considered. The other two strategies (i.e., *Least Misery* and *Most Pleasure*) are usually employed to model small groups; indeed, if a group is large, the group model would contain respectively only low or high ratings, which would not represent the preferences of the group as a whole.

Lastly, in case a person that guides the group or whose preferences align with most of the group exists, the *Most Respected Person* strategy would be at the same time effective and efficient to employ.

---

[1] Given 60 users who expressed a very low rating (like 2) for an item, a 64 bit machine would not be able to handle the group rating, since it cannot process numbers higher that 2^{52}.

4 Conclusions

In this paper, we analyzed the existing group modeling strategies and presented criteria for applicable design in real-world scenarios characterized by big data. As a result of this study, we can say that the vast majority of the strategies do not present limitations from an algorithmic point of view and could be efficiently implemented. However, due to how the strategies operate, their effectiveness is affected in big data scenarios. Indeed, some of them do not consider a group rating if a user or a part of the group has given a low rating to an item, or some others would lead the group model to be composed just with low or high ratings, blurring the knowledge on what the users in the group like or do not like. In conclusion, in presence of big data, a simple but very effective strategy, like *Additive Utilitarian*, which considers all the users and all the ratings given by them, should be preferred. Future work will be devoted at experimenting these strategies in the real-world scenarios characterized by big data to analyze their applicability and validate these design criteria.

References

1. 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. Volume 324 of Studies in Computational Intelligence. Springer Berlin Heidelberg (2011) 1–20
2. Jameson, A., Smyth, B.: Recommendation to groups. In: The Adaptive Web, Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science. Springer, Berlin (2007) 596–627
3. Yankelovich, D., Meer, D.: Rediscovering market segmentation. Harvard Business Review **84**(2) (2006) 1–10
4. Liu, Y., Kiang, M., Brusco, M.: A unified framework for market segmentation and its applications. Expert Syst. Appl. **39**(11) (September 2012) 10292–10302
5. Masthoff, J.: Group recommender systems: Combining individual models. In: Recommender Systems Handbook. Springer (2011) 677–702
6. Pizzutilo, S., Carolis, B.D., Cozzolongo, G., Ambruoso, F.: Group modeling in a public space: Methods, techniques and experiences. In: Proceedings of WSEAS AIC 05, Malta, ACM (2005)
7. Pessemier, T., Dooms, S., Martens, L.: Comparison of group recommendation algorithms. Multimedia Tools and Applications (2013) 1–45
8. McCarthy, J.F.: 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, Minneapolis (2002)
9. Christensen, I.A., Schiaffino, S.N.: Entertainment recommender systems for group of users. Expert Systems with Applications **38**(11) (2011) 14127–14135
10. Baltrunas, L., Makcinskas, T., Ricci, F.: Group recommendations with rank aggregation and collaborative filtering. In: Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, New York, NY, USA, ACM (2010) 119–126
11. Felfernig, A., Zehentner, C., Ninaus, G., Grabner, H., Maalej, W., Pagano, D., Weninger, L., Reinfrank, F.: Group decision support for requirements negotiation. In: Advances in User Modeling - UMAP 2011 Workshops, Revised Selected Papers. Volume 7138 of Lecture Notes in Computer Science., Springer (2012) 105–116

12. Senot, C., Kostadinov, D., Bouzid, M., Picault, J., Aghasaryan, A.: Evaluation of group profiling strategies. In: IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, IJCAI/AAAI (2011) 2728–2733
13. Senot, C., Kostadinov, D., Bouzid, M., Picault, J., Aghasaryan, A., Bernier, C.: Analysis of strategies for building group profiles. In: User Modeling, Adaptation, and Personalization, 18th International Conference, UMAP 2010. Proceedings. Volume 6075 of Lecture Notes in Computer Science., Springer (2010) 40–51
14. Lieberman, H., Dyke, N.W.V., Vivacqua, A.S.: Let's browse: A collaborative web browsing agent. In: IUI. (1999) 65–68
15. Bourke, S., McCarthy, K., Smyth, B.: Using social ties in group recommendation. In: AICS 2011: Proceedings of the 22nd Irish Conference on Artificial Intelligence and Cognitive Science: 31 August-2 September, 2011: University of Ulster-Magee, Intelligent Systems Research Centre (2011)
16. O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J.: Polylens: A recommender system for groups of users. In: Proceedings of the Seventh European Conference on Computer Supported Cooperative Work, Kluwer (2001) 199–218
17. Sánchez, L.Q., Bridge, D.G., Díaz-Agudo, B., Recio-García, J.A.: A case-based solution to the cold-start problem in group recommenders. In: Case-Based Reasoning Research and Development - 20th International Conference, ICCBR 2012. Proceedings. Volume 7466 of Lecture Notes in Computer Science., Springer (2012) 342–356
18. McCarthy, J.F., Anagnost, T.D.: Musicfx: An arbiter of group preferences for computer supported collaborative workouts. In: CSCW '98, Proceedings of the ACM 1998 Conference on Computer Supported Cooperative Work, ACM (1998) 363–372
19. Ardissono, L., Goy, A., Petrone, G., Segnan, M., Torasso, P.: Intrigue: Personalized recommendation of tourist attractions for desktop and hand held devices. Applied Artificial Intelligence **17**(8-9) (2003) 687–714
20. 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
21. Carolis, B.D., Pizzutilo, S.: Providing relevant background information in smart environments. In: E-Commerce and Web Technologies, 10th International Conference, EC-Web 2009. Proceedings. Volume 5692 of Lecture Notes in Computer Science., Springer (2009) 360–371
22. Martinez-Palau, X., Dominguez-Sal, D., Larriba-Pey, J.L.: Two-way replacement selection. Proc. VLDB Endow. **3**(1-2) (September 2010) 871–881
23. 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