Elsevier Editorial System(tm) for Decision Support Systems
Manuscript Draft

Manuscript Number: DECSUP-D-16-00019R3

Title: An empirical study of natural noise management in group recommendation systems

Article Type: Research Paper

Keywords: Recommender Systems
Group Recommender Systems
Natural Noise
Collaborative Filtering

Corresponding Author: Dr. Luis Martinez, PhD

Corresponding Author's Institution: University of Jaén

First Author: Jorge Castro

Order of Authors: Jorge Castro; Raciel Yera, PhD; Luis Martinez, PhD

Abstract: Group recommender systems (GRSs) filter relevant items to groups of users in overloaded search spaces using information about their preferences. When the feedback is explicitly given by the users, inconsistencies may be introduced due to various factors, known as natural noise. Previous research on individual recommendation has demonstrated that natural noise negatively influences the recommendation accuracy, whilst it improves when noise is managed. GRSs also employ explicit ratings given by several users as ground truth, hence the recommendation process is also affected by natural noise. However, the natural noise problem has not been addressed on GRSs. The aim of this paper is to develop and test a model to diminish its negative effect in GRSs. A case study will evaluate the results of different approaches, showing that managing the natural noise at different rating levels reduces prediction error. Eventually, the deployment of a GRS with natural noise management is analysed.

Jaén, September 25th, 2016

Prof. James R. Marsden
University of Connecticut, Storrs, Connecticut
USA

Dear Prof. James R. Marsden,

According to your last email, please find attached a copy of the revised version of our paper, as a possible publication for the Journal Decision Support Systems:

An empirical study of natural noise management in group recommendation systems. (DECSUP-D-16-00019)

The minor comments and suggestions pointed out for revision are reflected in the current version of the paper.

We enclose a detailed response regarding the modifications made in the paper according to the reviewers' comments.

I am looking forward to hearing from you.

Yours sincerely,

Luis Martínez
Computing School, Dept. Computer Sciences
Campus Las Lagunillas s/n, University of Jaén
23071 Jaén, Spain
Email: martin@ujaen.es

RESPONSE LETTER

First of all we would like to thank again the reviewers for their time and effort to review our paper and their valuable suggestions, which have helped us in improving the quality of our research.

In this revised version, we have attempted to address the minor issues identified by the reviewer # 1 that are detailed in the next pages.

Eventually, we believe that we have successfully completed the necessary additional work required on the paper.

<u>Reviewer #1:</u>

Authors have put serious efforts for incorporating the suggestions given in the previous round. In general, I am fine with the improvement of the work but I suggest few minor points:

If authors are making hypothesis, then the logic behind the statements given in the hypothesis has to be discussed before making the hypothesis for example:

H1: managing the natural noise only in the group ratings is enough to obtain improvements on a given GRS.

How the authors conceptualized that the "managing the natural noise only in group ratings is enough to obtain improvements on a given GRS" this argument needs to be built before the hypothesis, which is missing right now.

If not then the wording of the hypothesis should be changed or instead of making a hypothesis just make research questions and do not pass any judgement right now.

The paper has been modified to include the rationale behind ***each hypothesis*** according to referee's suggestion, before starting each of them. The modifications before each hypothesis are the below ones:

> Taking into account that group members' ratings are key data in GRSs for computing the recommendations provided to the group by their aggregation [11], their quality influences the recommendation accuracy. It leads to the formulation of the first hypothesis:
> …
>
> On the other hand, NNM for individual RSs has shown clear improvements [29]. Due to the fact that GRSs approaches are supported by individual RSs [11], a new hypothesis is raised:…
> …
>
> Eventually, if both levels (group ratings, all users ratings) are considered, some ratings tagged as noisy at the group level could be not noisy at the global level, and viceversa. Therefore, the NNM at both levels could lead to a better recommendation accuracy. Hence, H3 is formulated as follows:…

Besides that, I also suggest to further strengthen the argument given for the comment 2 (How this problem is different from single recommendation noise management in terms of methodological contribution).

A paragraph to highlight the methodological contributions in terms of NNM to GRS has been added:

> In group recommendation, unlike traditional recommendation scenarios, the ratings dataset has different levels of information from the group viewpoint. Therefore, the direct application of NNM methodologies is not adequate in this context [29]. This fact makes necessary to develop *new methodologies* that enable the application of NNM across the different levels of information in GRSs. This is the objective of the current paper, which will be reached by the study of different hypotheses.

Reviewer #2:

Good job. I look forward to seeing this work in DSS.

Thank you very much for your positive view and paper acceptance

HIGHLIGHTS FOR REVIEW

- The rationale behind ***each hypothesis*** has been included before expressing them.
- The methodological contributions for applying Natural Noise Management to GRSs have been highlighted.

An empirical study of natural noise management in group recommendation systems

Jorge Castro[a,*], Raciel Yera[b], Luis Martínez[c]

[a] Department of Computer Science and Artificial Intelligence, University of Granada, Granada (Spain)
[b] University of Ciego de Ávila, Ciego de Ávila (Cuba)
[c] Computer Science Department, University of Jaén, Jaén (Spain)

Abstract

Group recommender systems (GRSs) filter relevant items to groups of users in overloaded search spaces using information about their preferences. When the feedback is explicitly given by the users, inconsistencies may be introduced due to various factors, known as natural noise. Previous research on individual recommendation has demonstrated that natural noise negatively influences the recommendation accuracy, whilst it improves when noise is managed. GRSs also employ explicit ratings given by several users as ground truth, hence the recommendation process is also affected by natural noise. However, the natural noise problem has not been addressed on GRSs. The aim of this paper is to develop and test a model to diminish its negative effect in GRSs. A case study will evaluate the results of different approaches, showing that managing the natural noise at different rating levels reduces prediction error. Eventually, the deployment of a GRS with natural noise management is analysed.

Keywords: `group recommender systems`, `natural noise`, `collaborative filtering`

*Corresponding author

Email addresses: `jcastro@decsai.ugr.es` (Jorge Castro), `yeratoledo@gmail.com` (Raciel Yera), `martin@ujaen.es` (Luis Martínez)

1. Introduction

A recommender system (RS) focuses on providing personalised access to items (or information in general) in an overloaded search space. With this purpose in mind, RSs explore users' interests to find out which items are the most suitable. RSs have been successfully applied to support users trying to overcome the information overload problem in several domains [15], such as e-commerce [7], financial investment [20], e-learning [16, 28], e-government [14], and e-tourism [22], among others.

RSs employ different approaches depending on the information they rely on, but the most widespread ones are content-based and collaborative filtering approaches [1]. A content-based recommender system (CBRS) relies on users' preferences and information about items, such as features or a textual description, to be able to make recommendations [10]. On the other hand, collaborative filtering recommender systems (CFRSs) rely on users' preferences to be able to make recommendations [21]. Therefore, CFRSs are able to generate effective recommendations using only users' preferences (rating values) [25] hence the quality of the ratings influences the quality of the recommendations.

The ratings can be gathered implicitly [6] or explicitly, this article focuses on the latter case. Prior research has explored how noisy preferences intentionally inserted by users affect RSs [8, 13], so-called *malicious noise*. However, the noisy ratings introduced unintentionally by users, so-called *natural noise*, has recently attracted the attention of researchers. Several proposals investigate the detection and correction of such natural noise. Some proposals perform these tasks by exploiting the items' attributes [24], either by taking advantage of user's interaction [3], or by using knowledge extracted from the ratings themselves [29]. The main benefit of these proposals is their positive impact on the recommendations.

So far, natural noise has been studied only in RSs for individuals. However, Group Recommender Systems (GRSs) [11] play an important role in many social activities that require recommendations to be delivered to a group of users, such as watching TV with family, sightseeing with others, or going to the cinema with friends.

GRS approaches extend individual RS for recommending to groups by aggregating user individual information [18]. Therefore, GRSs use explicit ratings, and natural noise is also present biasing the group recommendation. Consequently, its management may play an important role in the quality of the group recommendations. This paper aims at researching the natural noise management (NNM) in GRS to study its influence in group recommendation.

In group recommendation, unlike traditional recommendation scenarios, the ratings dataset has different levels of information from the group viewpoint. Therefore, the direct application of NNM methodologies is not adequate in this context [29]. This fact makes necessary to develop new methodologies that enable the application of NNM across the different levels of information in GRSs. This is the objective of the current paper, which will be reached by the study of different hypotheses.

Taking into account that group members' ratings are key data in GRSs for computing the recommendations provided to the group by their aggregation [11], their quality influences the recommendation accuracy. It leads to the formulation of the first hypothesis:

- **H1**: NNM using only the group ratings would improve the group recommendation.

On the other hand, NNM for individual RSs has shown clear improvements [29]. Due to the fact that, GRSs approaches are supported by individual RSs [11], a new hypothesis is raised:

- **H2**: NNM in the entire ratings database, disregarding the groups, would improve the group recommendation.

Eventually, if both levels (group ratings, all users ratings) are considered, some ratings tagged as noisy at the group level could be not noisy at the global level, and viceversa. Therefore, the NNM at both levels could lead to a better recommendation accuracy. Hence, H3 is formulated as follows:

- **H3**: managing natural noise in the entire ratings database and, after that, adding a second step that manages natural noise in the group ratings, would improve the results as compared to a single step of NNM.

Different approaches which apply these NNM processes to GRSs are presented and a case study to verify the hypotheses is performed.

The remaining of the paper is organised as follows. Section 2 provides a background on CFRSs, GRSs, and natural noise. Section 3 presents four methods of natural noise management for GRSs. Section 4 develops the case study and analyses the results of the proposals, leading to the acceptance of H2 and H3. Finally, Section 5 points out the conclusions and discusses future research.

2. Background

This section briefly reviews the basics on CFRSs and GRSs. Eventually, some research work on NNM is discussed.

2.1. Collaborative Filtering Recommender Systems

Among the different approaches in RSs, such as CBRSs [9] or knowledge-based RSs [17], CFRSs are the most widespread approach because of their ability to generate good recommendations by using only users' ratings on the items.

Formally, the recommendation in CFRS is defined as the item (or set of items) that maximises the rating prediction for the target user u_j (see notation in Appendix A):

$$Recommendation(I, u_j) = arg \max_{i_k \in I} Prediction(u_j, i_k) \tag{1}$$

CFRS techniques can be mainly divided into two classes: memory-based and model-based methods [1]. Memory-based methods [21] employ the whole dataset in order to predict the rating values that the target user has not yet stated, and the item (or set of items) with the highest prediction is recommended. On the other hand, model-based methods use the ratings to learn a model that generates the recommendations. The model is built using machine learning techniques [4], such as matrix factorization, Bayesian networks, etc.

Due to the fact that CFRSs present certain limitations, these have been overcome by hybridization with other techniques using different algorithms [1].

Since the 1990s a large number of CF methods have been proposed. Two widely-used and effective methods are:

- *Resnick's user-based collaborative filtering* [26] predicts users' ratings for unexperienced items using users with similar preferences (neighbours).
- *Sarwar's item-based collaborative filtering* [27] predicts users' ratings combining the user's ratings for the neighbourhood of the target item.

Even though these methods are not the most recent, their effectiveness and simplicity have encouraged their use in many real-world systems. Furthermore, their properties have been extensively researched [21]. For these reasons, they are used in this paper as the single-user RS required as a part of the GRS.

2.2. Group Recommender Systems

Group recommendation is currently a research area with increasing importance because of the diversity of scenarios in which it is useful. To formalize the GRS problem, the notation in Appendix A is used. Group recommendation is commonly defined as the item (or set of items) that maximises the rating prediction for a group of users, G_a:

$$Recommendation(I, G_a) = arg\max_{i_k \in I} Prediction(i_k, G_a) \tag{2}$$

There are two basic approaches for group recommendation [11], based on single user recommendation:

- *Rating aggregation* (see Fig. 1): a pseudo-user is created for the group by aggregating members ratings. The recommendations are generated using this rating profile as the CFRS target user.

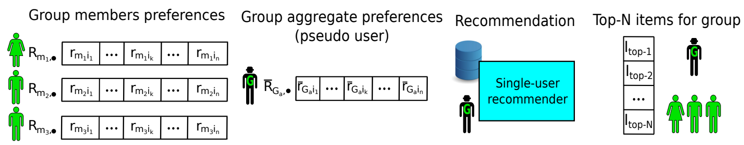


Figure 1: Rating aggregation approach for group recommendation.

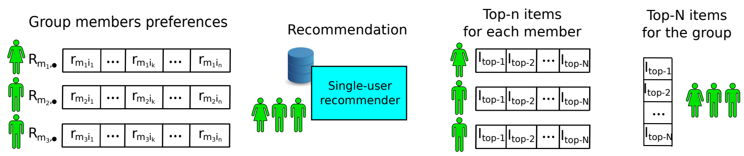


Figure 2: Recommendation aggregation approach for group recommendation.

- *Recommendation aggregation* (see Fig. 2): the members' individual recommendations are generated by a CFRS. The GRS aggregates them to produce a recommendation list targeted to the group.

In all scenarios, neither rating aggregation nor recommendation aggregation outperforms the other [18]. So, in order to identify the best both should be evaluated.

2.3. Natural noise

The management of unintentionally inconsistent user preferences, so called natural noise, is a relatively new research field in CFRSs [2, 24]. Natural noise appears due to factors such as the change of preferences over time, individual conditions, rating strategies, and social influences [24]. Amatriain et al. [2] developed a study to verify that natural noise biases the recommendations. The results show that the users tend to change their preferences, and that these inconsistencies could affect the recommendation accuracy.

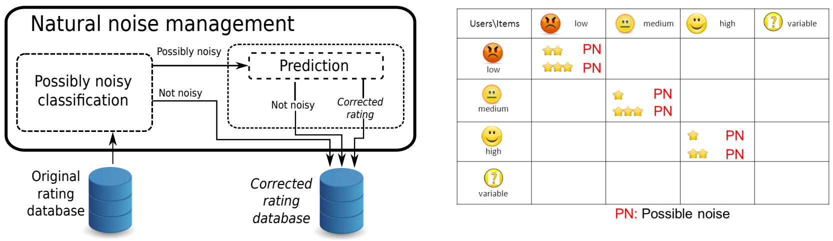
O'Mahony et al. [23] was the first research work to use the term natural noise. The authors suggest that if a rating r_{ui} is noise-free or if it contains natural noise, then it should be retained in the former case and removed in the latter. With this purpose, they establish the consistency between the original rating value and a new value predicted by using a recommendation algorithm for the same user-item pair. Later on, several authors focused on natural noise from different points of view [3, 24].

Previous research is based on different principles, but presents limitations, such as the removal of information from the dataset [23], or the need of additional information [3, 24]. To overcome these limitations, recently Yera et al. [29] proposed a two-step method that requires only the rating matrix (see Fig. 3a):

1. Noise detection: Ratings are tagged as *not noisy* or *possibly noisy* regarding their corresponding user and item behaviour (see Fig. 3b). Each rating and its corresponding user and item are classified as *high*, *medium* or *low*. If the user and item behaviour are the same and contradict rating classification, then the rating is tagged as possible noise.

2. Noise correction: For each possibly noisy rating a prediction is computed for its corresponding user and item. If the difference between the old and new value exceeds a threshold, then the prediction replaces the original value.

This NNM approach will be used in our proposal for GRS because it only needs the information in the rating matrix (further details in [29]). A simple example of its performance is shown based on the ratings shown in Table 1. The user and item classes (c_{u_j} and c_{i_k}, respectively) are the classes with the majority of ratings, defined as low=1,2 med=3 and high=4,5. If the majority is not absolute, then the class is variable. In the example, considering users' behaviour c_{u_j} and item tendency $R_{\bullet i_k}$, the ratings



(a) General scheme. (b) Classification of the ratings.

Figure 3: General scheme of natural noise management for individuals [29].

classified as possibly noisy are r_{u_1,i_3}, r_{u_3,i_3}, and r_{u_6,i_1} because they contradict the user behaviour and item tendency.

Table 1: Illustrative example for the classification of the ratings as possibly noisy.

	U						c_{i_k}	
	G_a							
	u_1	u_2	u_3	u_4	u_5	u_6	$R_{\bullet i_k}$	$R_{G_a i_k}$
i_1	5	5	5	5	5	2	high	high
i_2	5	3	5	3	3	3	medium	high
i_3	3	5	3	5	5	5	high	medium
i_4	5	5	5	5	5	5	high	high
i_5	1	1	4	2	1	5	low	low
c_{u_j}	high	high	high	high	high	high		

3. Natural noise management in group recommendation

The NNM is particularly interesting in GRSs, because it is not clear whether the natural noise of the members' ratings also bias the group recommendation. Therefore, it is important to verify that the NNM also plays an important role in GRS accuracy. However, NNM in individual RSs cannot be directly applied to GRSs, because of the proper features of GRSs. Therefore, different alternatives for NNM in group recommendation are introduced in this section.

To propose a NNM approach in GRS, in the group recommendation scenario two levels of data are considered to exist: (i) *local level*: preferences belonging to the group members, and (ii) *global level*: preferences of all the users in the entire dataset. The level considered most suitable to perform the NNM should then be studied. Following this aim four alternatives for NNM are then presented in both levels of data:

- First, two approaches that focus the NNM on the local level before the recommendations. The approaches are local NNM based on local information, NNM-LL, and local NNM based on global information, NNM-LG.
- Second, an approach that focuses the NNM on the global level before the recommendation, disregarding the group to which each user might belong. The approach is noted as global NNM approach (NNM-GG).
- Eventually, a cascade hybridization of the previous approaches is presented (NNM-H). It performs a global NNM approach, and then a local NNM that corrects the group ratings by using the information already corrected.

In the remainder of this section we use the notation introduced in Appendix A to detail the performance of each NNM approach.

3.1. Local natural noise management based on local information (NNM-LL)

NNM-LL approach is depicted in Figure 4a). It analyses the ratings of the target group, G_a, and corrects them by using only the information provided by the group members. This approach assumes that only the preferences associated with the group members should be taken into account in such characterisation, and thus this reduced amount of information might be enough to apply NNM in group recommendation.

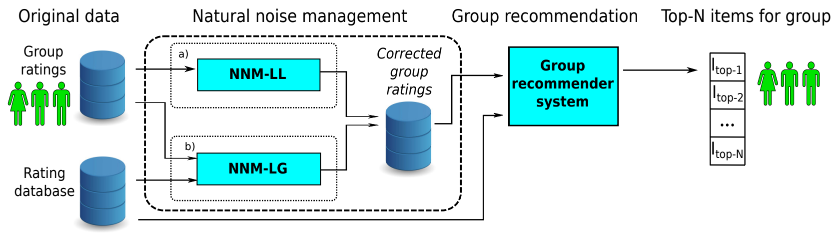


Figure 4: Scheme of local NNM for GRS, showing two approaches: a) NNM-LL, b) NNM-LG.

Algorithm 1 computes group recommendation by managing natural noise for each group G_a, using all the ratings of the group $R_{G_a,\bullet}$. *Algorithm* 2 adapts the NNM process [29] to group recommendation, i.e., NNM-LL, whose main feature is that the item classification uses only G_a local information, R_{G_a,i_k} (Algorithm 2, line 4).

Data: U,I,R,G
BuildRecommendationModel(U,I,R)
foreach G_a *in* G **do**
    $R^*_{G_a,\bullet}$ = NNMLL(G_a,$R_{G_a,\bullet}$)
    $recommendations_{G_a}$ = Recommend(G_a,$R^*_{G_a,\bullet}$)
return R^*

Algorithm 1: GRS with local NNM based on local information (NNM-LL).

Data: G_a,$R_{G_a,\bullet}$
Result: $R^*_{G_a,\bullet}$
1 **foreach** r_{m_l,i_k} *in* $R_{G_a,\bullet}$ **do**
2     $c_{r_{m_l,i_k}}$ = Classify(r_{m_l,i_k})
3   c_{m_l} = Classify(m_l, $R_{m_l,\bullet}$)
4   c_{i_k} = Classify(i_k, R_{G_a,i_k})
5   **if** $(c_{m_l} = c_{i_k})$ *and* $(c_{m_l} \neq c_{r_{m_l,i_k}})$ *and* $(c_{m_l} \neq variable)$ **then**
6         $r^*_{m_l,i_k}$ = Predict(R,m_l,i_k)
7   **else**
8         $r^*_{m_l,i_k} = r_{m_l,i_k}$
9 **return** $R^*_{G_a,\bullet}$

Algorithm 2: Procedure for local NNM based on local information (NNM-LL)

In NNM-LL only the data from the group G_a is taken into account. The small amount of data used to perform this analysis and correction makes it suitable to be applied online, when the recommendations are requested.

3.2. Local natural noise management based on global information (NNM-LG)

NNM-LG approach is depicted in Figure 4b). NNM-LG is similar to the previous approach, in terms of the rating set that it analyses and corrects. However, NNM-LG assumes that the ratings in G_a are not enough to be able to properly classify the items. Therefore, it classifies the items using all the information in the dataset, $R_{\bullet,i_k}$ (see Table 1).

A GRS with the NNM-LG approach follows the general scheme of *Algorithm* 1. However, for the item classification it uses all the ratings for the item, $R_{\bullet,i_k}$ (modifying Algorithm 2, line 4). This change modifies the item classification.

With NNM-LG the item profiling uses more information than the NNM-LL approach. This feature is key in recommendations targeted at groups because only a few ratings might be available for a given item, which might lead to a different classification of the items.

3.3. Global natural noise management (NNM-GG)

NNM-GG approach is depicted in Figure 5. It computes the recommendations by applying the NNM to the entire dataset (see *Algorithm* 3). All the ratings in the database are analysed and corrected, similarly to the NNM applied to single user RSs [29]. *Algorithm* 4 presents how the NNM-GG approach is applied to the entire dataset.

Due to the number of ratings being revised, the NNM-GG approach must be applied offline. However, it might result in a better NNM, because the recommendation model

Data: U,I,R,G
R^* = NNMGG(R)
BuildRecommendationModel(U,I,R^*)
foreach G_a *in* G **do**
  $recommendations_{G_a}$ = Recommend(G_a, $R^*_{G_a}$)

Algorithm 3: GRS with global natural noise management.

Data: R
Result: R^*
foreach r_{u_j,i_k} *in* R **do**
  $c_{r_{u_j,i_k}}$ = Classify(r_{u_j,i_k})
  c_{u_j} = Classify(u_j, $R_{u_j,\bullet}$)
  c_{i_k} = Classify(i_k, $R_{\bullet,i_k}$)
  if $(c_{u_j} = c_{i_k})$ *and* $(c_{u_j} \neq c_{r_{u_j,i_k}})$ *and* $(c_{u_j} \neq variable)$ **then**
    $r^*_{u_j,i_k}$ = Predict(R,u_j,i_k)
  else
    $r^*_{u_j,i_k} = r_{u_j,i_k}$
return R^*

Algorithm 4: Procedure for global natural noise management (NNM-GG)

is built with a preprocessed database, thus the natural noise influence in the model might be reduced.

3.4. Hybrid global-local natural noise management (NNM-H)

NNM-H approach is depicted in Figure 6. This approach combines NNM-GG and NNM-LG. The reason for this hybridisation is that once the NNM-GG is applied, the

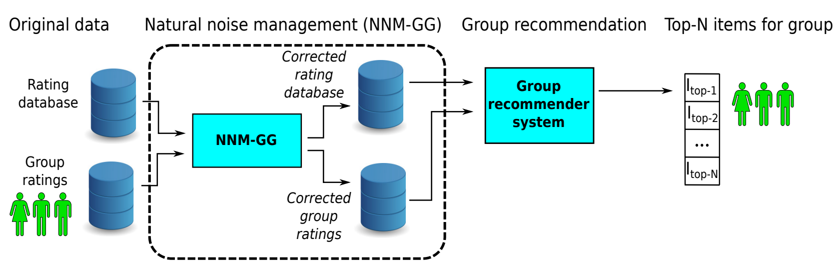


Figure 5: Global natural noise management based on global information (NNM-GG) application.

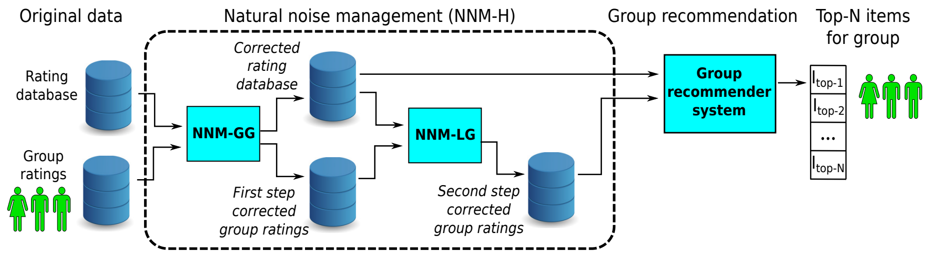


Figure 6: Hybrid global-local natural noise management (NNM-H) application.

classification of the group ratings and their correction might be affected by the modifications made in the initial dataset. Therefore, NNM-LG is then applied to revise the group ratings using the corrected ratings dataset.

Algorithm 5 presents NNM-H in which the entire dataset is initially corrected (line 1), and a local correction is then performed on the group ratings (line 4) to analyse the ratings regarding the revised dataset.

Data: U,I,R,G
1 R^{*} = NNMGG(R)
2 BuildRecommendationModel(U,I,R^{*})
3 **foreach** G_a *in* G **do**
4 $\quad R^{**}_{G_a,\bullet}$ = NNMLG(G_a,$R^{*}_{G_a,\bullet}$)
5 $\quad recommendations_{G_a}$ = Recommend(G_a,$R^{**}_{G_a,\bullet}$)

Algorithm 5: GRS with hybrid natural noise management (NNM-H)

3.5. Illustrative example

This section presents an illustrative example of the different approaches for NNM in GRS. With this aim in mind, the example uses the data shown in Table 1, in which the target group G_a is composed of users u_1, u_2, and u_3.

For NNM-LL only the ratings of the group members are evaluated. Each item is classified using the group ratings, R_{G_a,i_k}. Therefore, the possibly noisy rating is r_{u_2,i_2}.

For NNM-LG the rating set revised is also R_{G_a,i_k}, but the item classification is done with the complete dataset, $R_{\bullet,i_k}$. The possible noisy ratings are r_{u_1,i_3}, and r_{u_3,i_3}.

For NNM-GG, all the ratings in the database are revised, and the item classification is done with the complete dataset, $R_{\bullet,i_k}$. The possible noisy ratings are r_{u_1,i_3}, r_{u_3,i_3}, and r_{u_6,i_1}, which include the possible noisy detected by NNM-LG.

For NNM-H, the first step is to apply NNM-GG and correct the possibly noisy ratings, results are shown in Table 2, R^*. The second step is to apply NNM-LG over $R^*_{G_a,\bullet}$, considering the corrected ratings database. The possibly noisy rating of this second step is r_{u_3,i_3}.

Table 2: Rating database after the NNM-GG correction in Table 1. The values corrected by NNM-GG are highlighted in bold. This rating database is R^* and it is used as input for NNM-LG to produce $R^{**}_{G_a,\bullet}$.

	U						c_{i_k}
	G_a						
	u_1	u_2	u_3	u_4	u_5	u_6	$R_{\bullet i_k}$
i_1	5	5	5	5	5	**3**	high
i_2	5	3	5	3	3	3	medium
i_3	**4**	5	**3**	5	5	5	high
i_4	5	5	5	5	5	5	high
i_5	1	1	4	2	1	5	low
c_{u_j}	high	high	high	high	high	high	

4. Case study

To measure the effect of previous NNM approaches in GRSs performance, an experimental procedure is used to evaluate them. This section presents the experimental protocol used in the experiments, and the results are then presented and discussed to verify the hypotheses introduced in section 1.

4.1. Experimental protocol

This research applies a widely-used evaluation protocol [11] for GRSs, which is composed of four main steps: (i) compute the training and test partitions of the original dataset, (ii) generate the groups randomly, (iii) generate the group recommendation for each group using the training set, and (iv) individually evaluate the recommendations, as in single-user recommendation, by comparing the group recommendations with the users' ratings in the test set.

The design of a group recommendation algorithm must identify:

- The GRS approach which the algorithm uses.
- The aggregation scheme used to combine the members' information.
- The single-user RS that the GRS uses internally.

Section 2.2 introduced the group recommendation approaches to be evaluated: *rating aggregation* and *recommendation aggregation*. Because each approach manages the users' preferences in a different way, one could expect that the effect of the NNM algorithm in a GRS strongly depends on whether it is based on rating or recommendation aggregation. The effects of the NNM proposals (Section 3) on each GRS approach will be analysed separately.

Regarding the aggregation approaches, De Pessemier et al. [11] refer to different methods used to aggregate the preferences of the group. Several works [5, 12] have pointed out that the average (Avg) and least misery (Min) approaches obtain the best results. Therefore, these aggregation approaches are used in our evaluation.

Finally, the experimental procedure evaluates the NNM approaches presented, comparing them with baseline approaches which do not perform a NNM, and single user RS: the Resnick's user-based (UB) and the Sarwar's item-based (IB) CF methods (see Section 2.1.

The datasets used in this case study are:

- The MovieLens 100k dataset[1], composed of a 100.000 ratings given by 943 users over 1682 movies in the one to five stars domain.

- The Netflix Tiny dataset[2], composed of 4427 users, 1000 items, and 56136 ratings, in the same domain. Due to the high sparsity of this dataset we use only those users with more than 10 ratings. Specifically, 1757 users are used.

The case study is focused on evaluating the techniques in occasional groups, therefore the users are grouped randomly, as done in previous research on GRS [11].

Regarding the amount of available users in the dataset, each NNM approach (incorporating an aggregation approach and a single-user recommender) is evaluated using 50 randomly generated groups, and this procedure is repeated 20 times. The evaluation measure **MAE** is used for all cases.

The sizes of the groups are 5, 10 and 15 for each evaluation. Larger groups are excluded because they are not used in these kinds of experimental scenarios [11, 12].

Below, the performance of the NNM approaches described in section 3 are evaluated in a recommendation aggregation-based GRS with the mentioned aggregation

[1]collected by GroupLens Research Project at the University of Minnesota (http://grouplens.org)
[2]small version of Netflix dataset, in Personalised Recommendation Algorithms Toolkit (http://prea.gatech.edu)

Table 3: MAE results for the recommendation aggregation approach.

Dataset	Prediction Technique	Group size	Base	NNM-LL	NNM-LG	NNM-GG	NNM-H
MovieLens 100k	IB+Avg	5	0.8779	0.8778	0.8747	0.8607	**0.8583**
		10	0.8998	0.8996	0.8956	0.8837	**0.8804**
		15	0.9080	0.9079	0.9036	0.8913	**0.8875**
	IB+Min	5	1.0218	1.0214	1.0137	1.0045	**0.9983**
		10	1.1404	1.1403	1.1328	1.1263	**1.1200**
		15	1.2066	1.2066	1.1959	1.1918	**1.1830**
	UB+Avg	5	0.8053	0.8053	0.8051	0.7853	**0.7852**
		10	0.8127	0.8127	0.8125	0.7932	**0.7931**
		15	0.8146	0.8145	0.8145	**0.7946**	**0.7946**
	UB+Min	5	0.8421	0.8419	0.8399	0.8190	**0.8172**
		10	0.8700	0.8698	0.8674	0.8463	**0.8444**
		15	0.8845	0.8844	0.8815	0.8593	**0.8572**
Netflix Tiny	IB+Avg	5	0.8435	0.8431	0.8415	0.8386	**0.8368**
		10	0.8630	0.8627	0.8598	0.8572	**0.8542**
		15	0.8627	0.8626	0.8595	0.8569	**0.8539**
	IB+Min	5	1.0074	1.0072	1.0025	1.0011	**0.9963**
		10	1.1451	1.1445	1.1330	1.1364	**1.1262**
		15	1.2252	1.2251	1.2117	1.2169	**1.2046**
	UB+Avg	5	0.8127	0.8128	0.8130	**0.8060**	0.8062
		10	0.8245	0.8245	0.8244	0.8174	**0.8173**
		15	0.8194	0.8194	0.8193	**0.8129**	**0.8129**
	UB+Min	5	0.8616	0.8615	0.8609	0.8538	**0.8532**
		10	0.9034	0.9033	0.9020	0.8945	**0.8934**
		15	0.9192	0.9192	0.9187	0.9108	**0.9103**

schemes and single-user RSs. Then, Section 4.3 develops a similar study for rating aggregation-based GRS. Finally, Section 4.4 discusses the results.

4.2. Results in recommendation aggregation-based GRS

This section presents the results of the NNM for GRS based on recommendation aggregation. Table 3 shows the experimental results for MovieLens 100k and Netflix Tiny datasets, regarding the mentioned aggregation approaches (Avg and Min) and single-user RSs (IB and UB). For each case, the best NNM result for each GRS has been highlighted.

In general, the results clearly show that NNM leads to better performance of the group recommendation algorithms, but such improvement is closely associated with the NNM approach:

- **NNM-LL**. All cases show that the application of the NNM-LL approach does not imply a significant improvement in performance. Specifically, NNM-LL obtains a similar performance to that of the baseline. This therefore suggests that the use of local information is not enough to be able to properly characterise the items and manage the natural noise of the group information.

- **NNM-LG**. In contrast to NNM-LL, the NNM-LG approach, overall, produces a slight improvement as compared to the baseline method. For IB+Min the improvement observed is greater, specifically it is around 0.01 better. However, for UB+Avg it does not provide a significant improvement, since the MAE difference is less than 0.001. In the remaining cases the results for NNM-LG show a narrow improvement. These facts show that NNM-LG performs better than NNM-LL, which suggests that using more information to characterise the items provides improvements in the NNM for recommendation aggregation.

- **NNM-GG**. The results show that the NNM of the entire dataset outperforms the local correction. Considering that the natural noise is distributed across all the dataset, the NNM approaches described in section 3 were expected to cause an improvement in this scenario, as it has been demonstrated that the NNM in single-user recommendation introduces a performance improvement [29]. In general, NNM-GG obtains improvements when compared to the baseline and local approaches.

- **NNM-H**. The best performance for the evaluated cases are obtained by the NNM-

Table 4: MAE results for the rating aggregation approach.

Dataset	Prediction Technique	Group size	Base	NNM-LL	NNM-LG	NNM-GG	NNM-H
MovieLens 100k	IB+Avg	5	0.8664	0.8664	0.8657	0.8477	**0.8473**
		10	0.8898	0.8898	0.8884	0.8723	**0.8712**
		15	0.8990	0.8990	0.8969	0.8814	**0.8797**
	IB+Min	5	0.8877	0.8874	0.8808	0.8674	**0.8617**
		10	0.9563	0.9559	0.9387	0.9347	**0.9197**
		15	1.0252	1.0245	0.9942	1.0006	**0.9770**
	UB+Avg	5	0.7998	0.7997	0.7997	**0.7801**	0.7802
		10	0.8101	0.8101	0.8102	0.7913	**0.7912**
		15	0.8133	0.8133	0.8135	**0.7937**	0.7938
	UB+Min	5	0.8019	0.8017	0.8013	0.7820	**0.7818**
		10	0.8139	0.8138	0.8136	0.7944	**0.7943**
		15	0.8188	0.8188	0.8186	**0.7983**	**0.7983**
Netflix Tiny	IB+Avg	5	0.8382	0.8381	0.8376	0.8330	**0.8325**
		10	0.8637	0.8638	0.8624	0.8579	**0.8566**
		15	0.8682	0.8682	0.8663	0.8621	**0.8604**
	IB+Min	5	0.8689	0.8683	0.8643	0.8628	**0.8585**
		10	0.9351	0.9344	0.9234	0.9282	**0.9163**
		15	0.9809	0.9801	0.9630	0.9738	**0.9564**
	UB+Avg	5	0.8092	0.8094	0.8092	**0.8025**	0.8027
		10	0.8221	0.8223	0.8222	**0.8152**	0.8153
		15	0.8178	0.8178	0.8178	0.8111	**0.8110**
	UB+Min	5	0.8143	0.8143	0.8134	0.8075	**0.8067**
		10	0.8267	0.8269	0.8265	0.8203	**0.8198**
		15	0.8226	0.8226	0.8223	0.8162	**0.8158**

H approach. The results show that the hybridisation of NNM-GG and NNM-LG outperforms their own individual performance. NNM-H obtains the best results for the IB prediction technique with both Min and Avg aggregations, and for UB+Min.

4.3. Results in rating aggregation-based GRS

This section presents the results of the NNM approaches applied to rating aggregation-based GRS. Similarly to the previous section, the results are presented in Table 4, regarding the aggregation approaches (Avg and Min) and single-user RS (IB and UB). The best NNM result for each configuration has been highlighted.

In general, the results show a similar performance of the NNM to the previous GRS approach. The improvements that each NNM approach provides as compared to the baseline are different, and thus the results of each technique are analysed separately:

- **NNM-LL**. The results do not show improvements as compared to the baseline. Therefore, it again suggests that the use of local information is not enough to properly characterise the items and manage the natural noise of the group information.

- **NNM-LG**. In general, the NNM-LG approach provides very narrow improvements for the baseline. Specifically, it only provides improvements for the IB+Min GRS approach for big groups in both datasets. This improvement might be due to the differences in MAE which the baseline obtains for IB+Min and IB+Avg, thus the NNM approach has a greater margin for improvement.

- **NNM-GG**. The NNM-GG approach improves the accuracy of the baseline and the local approaches, NNM-LL and NNM-LG. This behaviour might be due to the amount of ratings being analysed and corrected.

- **NNM-H**. Overall, the results of the NNM-H approach place it as the best NNM approach for rating aggregation.

4.4. Discussion

This section focuses on the verification of the hypotheses presented in Section 1 by analysing the experimental results. The results obtained determine that **H1 is rejected**, managing the natural noise only in the group ratings is enough to obtain improvements on a given GRS. On the other hand, **H2 is accepted**, managing the natural noise in the

Table 5: Paired samples t-test p-values to compare each natural noise management technique with the baseline on MovieLens 100k dataset.

Dataset	Group aggregation	Prediction technique	Group size	NNM-LL	NNM-LG	NNM-GG	NNM-H
MovieLens 100k	Rating	IB+Avg	5	0.1573	<**0.0001**	<**0.0001**	<**0.0001**
			10	0.1770	<**0.0001**	<**0.0001**	<**0.0001**
			15	0.6921	<**0.0001**	<**0.0001**	<**0.0001**
		IB+Min	5	<**0.0001**	<**0.0001**	<**0.0001**	<**0.0001**
			10	<**0.0001**	<**0.0001**	<**0.0001**	<**0.0001**
			15	<**0.0001**	<**0.0001**	<**0.0001**	<**0.0001**
		UB+Avg	5	0.9587	0.9225	<**0.0001**	<**0.0001**
			10	**0.0245**	0.2663	<**0.0001**	<**0.0001**
			15	0.1083	**0.0013**	<**0.0001**	<**0.0001**
		UB+Min	5	0.3497	0.0774	<**0.0001**	<**0.0001**
			10	0.1861	0.2058	<**0.0001**	<**0.0001**
			15	0.7174	0.0922	<**0.0001**	<**0.0001**
	Recommendation	IB+Avg	5	**0.0134**	<**0.0001**	<**0.0001**	<**0.0001**
			10	<**0.0001**	<**0.0001**	<**0.0001**	<**0.0001**
			15	**0.0005**	<**0.0001**	<**0.0001**	<**0.0001**
		IB+Min	5	**0.0005**	<**0.0001**	<**0.0001**	<**0.0001**
			10	0.2681	<**0.0001**	<**0.0001**	<**0.0001**
			15	0.1190	<**0.0001**	<**0.0001**	<**0.0001**
		UB+Avg	5	0.6047	0.1038	<**0.0001**	<**0.0001**
			10	0.9303	**0.0304**	<**0.0001**	<**0.0001**
			15	0.7342	**0.0096**	<**0.0001**	<**0.0001**
		UB+Min	5	**0.0275**	<**0.0001**	<**0.0001**	<**0.0001**
			10	**0.0003**	<**0.0001**	<**0.0001**	<**0.0001**
			15	<**0.0001**	<**0.0001**	<**0.0001**	<**0.0001**

entire ratings database, disregarding the groups, improves the group recommendation. Also, **H3 is accepted**, managing the natural noise in the entire ratings database and, after that, adding a second step that manages the natural noise in the group ratings, improves the results as compared to only applying a single step of NNM. The details of the analysis performed to test the hypotheses are explained in the following sections.

H1: Managing natural noise in the group ratings only would improve the GRS.

The H1 hypothesis has been tested by analysing the results of the local level approaches: NNM-LL and NNM-LG compared with the baseline. In general the local NNM provides narrow improvements, although for the rating aggregation GRS with

IB+Min there are improvements when comparing NNM-LG with the baseline. To test whether the results obtained are significant, the paired samples t-test is applied to the results of the executions. This test compares whether the differences found in the paired samples are statistically different. The test is applied to determine if the corresponding local NNM technique results are different to the baseline on each case. The p-values for each of the cases tested are depicted in Tables 5 and 6 for MovieLens 100k and Netflix Tiny datasets, respectively (NNM-LL and NNM-LG columns). The tests that were able to reject the equality with a confidence level of 95% have been highlighted. Although some of the results show statistical differences between the baseline and NNM-LL and NNM-LG, their improvements are limited to specific cases. In the case of NNM-LL, results improve for both datasets for rating aggregation with IB+Min and for recommendation aggregation with IB+Avg. In the case of NNM-LG, results improve for both datasets with the IB prediction technique.

Therefore, hypothesis **H1 is rejected** as it only shows statistically significant improvements in a few cases. Hence, the application of local based techniques does not provide significant improvements to the group recommendation.

H2: Managing natural noise in the entire ratings database, disregarding the groups, would improve the group recommendation.

The H2 hypothesis has been tested by analysing the results of NNM-GG and NNM-H approaches compared with the baseline. In general, the proposals improve the results of the baseline in terms of MAE by around 0.02. To test whether the results obtained are significant, the results are tested similarly to H1 and the p-values are depicted in Tables 5 and 6 for MovieLens and Netflix Tiny datasets, respectively (NNM-GG and

Table 6: Paired samples t-test p-values to compare each natural noise management technique with the baseline on Netflix Tiny dataset.

Dataset	Group aggregation	Prediction technique	Group size	NNM-LL	NNM-LG	NNM-GG	NNM-H
Netflix Tiny	Rating	IB+Avg	5	0.3459	**0.0104**	<**0.0001**	<**0.0001**
			10	0.3524	<**0.0001**	<**0.0001**	<**0.0001**
			15	0.1991	<**0.0001**	<**0.0001**	<**0.0001**
		IB+Min	5	**0.0040**	<**0.0001**	<**0.0001**	<**0.0001**
			10	<**0.0001**	<**0.0001**	<**0.0001**	<**0.0001**
			15	<**0.0001**	<**0.0001**	<**0.0001**	<**0.0001**
		UB+Avg	5	0.1121	0.9215	<**0.0001**	<**0.0001**
			10	0.1968	0.5745	<**0.0001**	<**0.0001**
			15	0.6876	0.9641	<**0.0001**	<**0.0001**
		UB+Min	5	0.6833	0.1541	<**0.0001**	<**0.0001**
			10	**0.0154**	0.6594	<**0.0001**	<**0.0001**
			15	0.7768	0.3047	<**0.0001**	<**0.0001**
	Recommendation	IB+Avg	5	**0.0112**	<**0.0001**	<**0.0001**	<**0.0001**
			10	**0.0002**	<**0.0001**	<**0.0001**	<**0.0001**
			15	**0.0002**	<**0.0001**	<**0.0001**	<**0.0001**
		IB+Min	5	0.2777	<**0.0001**	<**0.0001**	<**0.0001**
			10	0.0886	<**0.0001**	<**0.0001**	<**0.0001**
			15	0.3985	<**0.0001**	<**0.0001**	<**0.0001**
		UB+Avg	5	0.0635	0.1132	<**0.0001**	<**0.0001**
			10	0.9263	0.4457	<**0.0001**	<**0.0001**
			15	0.3817	0.3929	<**0.0001**	<**0.0001**
		UB+Min	5	0.8274	0.0894	<**0.0001**	<**0.0001**
			10	0.5147	**0.0033**	<**0.0001**	<**0.0001**
			15	0.4580	0.0997	<**0.0001**	<**0.0001**

NNM-H columns). The tests that were able to reject the equality with a confidence level of 95% have been highlighted. All the statistical tests that compare NNM-GG and NNM-H with the baseline reject the equality of the results. These results show that the global NNM approaches improve the group recommendation across different group aggregation and prediction techniques in both datasets evaluated.

Therefore, the hypothesis **H2 is accepted**. The greater improvement achieved by NNM-GG and NNM-H over the baseline may occur because group recommendation with CFRS depends not only on the group ratings, but also on the collaborative information that the remaining users provide. Therefore, NNM at global level improves the predictions, and thus the GRSs produce better recommendations.

Table 7: Paired samples t-test p-values to compare NNM-GG with NNM-H.

		Group aggregation			
		Rating		Recommendation	
Prediction technique	Group size	MovieLens 100k	Netflix Tiny	MovieLens 100k	Netflix Tiny
IB+Avg	5	**0.0013**	**0.0136**	<**0.0001**	<**0.0001**
	10	<**0.0001**	<**0.0001**	<**0.0001**	<**0.0001**
	15	<**0.0001**	<**0.0001**	<**0.0001**	<**0.0001**
IB+Min	5	<**0.0001**	<**0.0001**	<**0.0001**	<**0.0001**
	10	<**0.0001**	<**0.0001**	<**0.0001**	<**0.0001**
	15	<**0.0001**	<**0.0001**	<**0.0001**	<**0.0001**
UB+Avg	5	0.2662	0.5664	0.1273	0.4065
	10	0.3890	0.6813	0.1118	0.6998
	15	**0.0466**	0.6162	0.5110	0.4616
UB+Min	5	0.3643	0.2169	<**0.0001**	0.3520
	10	0.5637	0.3818	<**0.0001**	**0.0040**
	15	0.7230	0.1835	<**0.0001**	**0.0250**

H3: Managing natural noise in the entire ratings database and, after that, adding a second step that manages natural noise in the group ratings, would improve the results as compared to a single step of NNM.

The H3 hypothesis has been tested by comparing the results of NNM-GG with NNM-H. In general, the NNM-H approach improves the results of the NNM-GG approach. To test whether the improvements obtained are significant, the results have been statistically tested. The results of the paired samples t-test to compare NNM-GG and NNM-H on the different configurations are depicted in Table 7. The tests show that the techniques present statistically significant improvements in general.

Specifically, analysing the results in detail, it is clear that the NNM-H approach shows improvements compared to NNM-GG in all the configurations with the prediction technique IB, in both datasets and in both rating and recommendation aggregation. If we focus on the results for UB prediction technique, NNM+H provides improvements in recommendation aggregation with UB+Min.

Therefore, **H3 is accepted**, for IB prediction techniques and UB+min with recommendation aggregation. Consequently, the best approach for managing natural noise in group recommendation is the NNM-H approach overall with IB prediction technique.

4.5. Complexity and deployment

According to previous section NNM-H performs better than NNM-GG for IB prediction techniques. Hence we focus on the deployment of a real-world GRS with NNM-H and IB prediction, taking two important issues into account: (i) the complexity order of the NNM-H approach and (ii) the update frequency of the IB recommendation model. Consequently this section studies the complexity of the NNM-H approach and provides some advice on its deployment in a IB prediction. The notations are based on the introduced in Appendix A.

1. *Complexity order of NNM-H approach*

   The NNM-H approach consists of applying NNM-GG before the IB model building and applying NNM-LG before the recommendation phase. First the complexity order of the NNM-GG approach is studied, which is composed of the rating detection and the rating correction:

   - Rating detection: the ratings are evaluated to detect the noisy ones. The rating detection complexity order is $O(m \cdot p)$, being $p = max(|R_{u_j,\bullet}|)$.
   - Rating correction: a correction is computed for each noisy rating using UKNN. If the noisy ratings of each user are grouped to be corrected (optimisation over Algorithm 4), then the UKNN neighbour computation, which

is the costly part of the correction, is done once per user. Therefore the complexity of the rating correction for one user is $O(m \cdot p)$.

Given that $m \gg p$ [21], the complexity order of NNM-GG is $O(m^2)$. Now it is necessary to study the complexity of NNM-LG in a similar way:

- Rating detection: its complexity order depends on the number of ratings given by the group, therefore it is $O(r \cdot p)$, being $p = max(|R_{m_l,\bullet}|)$.
- Rating correction: the correction is done similarly to NNM-GG. The difference is the necessity of computing the UKNN neighbours once per group member (optimisation over Algorithm 2 for NNM-LG), therefore the rating correction complexity is $O(m \cdot p)$ for one group member.

Therefore the complexity order of NNM-LG is $O(r * (p + m * p))$. Given that $m \gg p$ [21], NNM-LG complexity order is $O(r \cdot m)$. Summarising, the complexity of NNM-H is $O(m^2)$ in the model computation and $O(r \cdot m)$ in the recommendation computation.

2. *Deployment of a GRS with NNM-H and IB prediction*

The GRSs based on IB prediction generate a model to compute the recommendations, which have a complexity order $O(n^3)$ [21]. In a deployed RS, the item based model is calculated offline and updated with certain frequency [30], typically daily. In this case, NNM-H integration is straightforward.

On the other hand, there are domains in which complete model updates are not affordable. This happens in domains that show a high data variation, such as *advertising* or news recommendation, or in domains where the volume of the data results in expensive model computation. Such problems have been addressed with incremental models [19], in which new ratings trigger partial updates of the model. In this case the NNM should also be applied incrementally. When a new rating r_{ui} is added, it might be classified as noise and therefore corrected. In the unusual case where this rating changes the user or item classification [29], all the ratings in the corresponding user or item columns should be checked. All these tasks could be performed with a low computational cost.

In short we can conclude that the advantages of NNM-H in GRSs clearly outweigh the time and computing costs it requires.

5. Concluding Remarks and Future Works

This paper presents four approaches to manage and correct the natural noise in group recommender systems. The results show that NNM over the group ratings provides slight improvements to the group recommendation performance. On the other hand, when the NNM is applied to the entire dataset, it clearly increases the performance of the group recommender systems. The best results were obtained by the NNM-H approach, which performs a cascade hybridisation of the global and local approaches, i.e., it manages the natural noise first at a global level (entire dataset) and then, manages the natural noise with the corrected database at a local level (group ratings). The results show clear improvements in the case study developed.

In the future we will explore the use of fuzzy tools to provide a better representation of the group information. With a better understanding of the group information, a more effective and flexible NNM can be performed, leading to improvements in the group recommendations. In addition, the role of NNM in cold-start recommendation scenarios will be explored.

Acknowledgements: This research work was partially supported by the Research Project TIN-2015-66524-P, the Spanish FPU fellowship (FPU13/01151), and also the Eureka SD Project (agreement number 2013-2591).

Bibliography

[1] Gediminas Adomavicius and Alexander Tuzhilin Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. Data Eng.*, 17(6):734–749, 2005.

[2] Xavier Amatriain, Josep M Pujol, and Nuria Oliver. I like it... i like it not: Evaluating user ratings noise in recommender systems. In *User modeling, adaptation, and personalization*, pages 247–258. Springer, 2009.

[3] Xavier Amatriain, Josep M Pujol, Nava Tintarev, and Nuria Oliver. Rate it again: increasing recommendation accuracy by user re-rating. In *Proceedings of the third ACM conference on Recommender systems*, pages 173–180. ACM, 2009.

[4] Xavier Amatriain and JosepM. Pujol. Data mining methods for recommender systems. In Francesco Ricci, Lior Rokach, and Bracha Shapira, editors, *Recommender Systems Handbook*, pages 227–262. Springer US, 2015.

[5] Sihem Amer-Yahia, Senjuti Basu Roy, Ashish Chawlat, Gautam Das, and Cong

Yu. Group recommendation: Semantics and efficiency. *Proceedings of the VLDB Endowment*, 2(1):754–765, 2009.

[6] Josef Bauer and Alexandros Nanopoulos. Recommender systems based on quantitative implicit customer feedback. *Decision Support Systems*, 68:77 – 88, 2014.

[7] J. Ben Schafer, Joseph A. Konstan, and John Riedl. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5(1-2):115–153, 2001.

[8] Robin Burke, MichaelP. O'Mahony, and NeilJ. Hurley. Robust collaborative recommendation. In Francesco Ricci, Lior Rokach, and Bracha Shapira, editors, *Recommender Systems Handbook*, pages 961–995. Springer US, 2015.

[9] Jorge Castro, Rosa M. Rodríguez, and Manuel J. Barranco. Weighting of features in content-based filtering with entropy and dependence measures. *International Journal of Computational Intelligence Systems*, 7(1):80–89, 2014.

[10] Marco de Gemmis, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro. Semantics-aware content-based recommender systems. In Francesco Ricci, Lior Rokach, and Bracha Shapira, editors, *Recommender Systems Handbook*, pages 119–159. Springer US, 2015.

[11] Toon De Pessemier, Simon Dooms, and Luc Martens. Comparison of group recommendation algorithms. *Multimed. Tools. Appl.*, 72(3):2497–2541, 2014.

[12] Pragya Dwivedi and Kamal K Bharadwaj. e-learning recommender system for a group of learners based on the unified learner profile approach. *Expert Systems*, 32:264 – 276, 2015.

[13] Ihsan Gunes, Cihan Kaleli, Alper Bilge, and Huseyin Polat. Shilling attacks against recommender systems: a comprehensive survey. *Artificial Intelligence Review*, 42(4):767–799, 2014.

[14] Xuetao Guo and Jie Lu. Intelligent e-government services with personalized recommendation techniques. *International Journal of Intelligent Systems*, 22(5):401–417, 2007.

[15] Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang, and Guangquan Zhang. Recommender system application developments: A survey. *Decision Support Systems*, 74:12–32, 2015.

[16] N. Manouselis, R. Vuorikari, and F. Van Assche. Collaborative recommendation of e-learning resources: an experimental investigation. *Journal of Computer Assisted Learning*, 26(4):227–242, 2010.

[17] Luis Martínez, Manuel J. Barranco, Luis G. Pérez, and Macarena Espinilla. A knowledge based recommender system with multigranular linguistic information. *International Journal of Computational Inteligence Systems*, 1(3):225–236, 2008.

[18] Judith Masthoff. Group recommender systems: Aggregation, satisfaction and group attributes. In Francesco Ricci, Lior Rokach, and Bracha Shapira, editors, *Recommender Systems Handbook*, pages 743–776. Springer US, 2015.

[19] Catarina Miranda and Alípio Mário Jorge. *Proceedings of Progress in Artificial Intelligence: 14th Portuguese Conference on Artificial Intelligence.*, chapter Item-Based and User-Based Incremental Collaborative Filtering for Web Recommendations, pages 673–684. Springer Berlin Heidelberg, 2009.

[20] Cataldo Musto, Giovanni Semeraro, Pasquale Lops, Marco de Gemmis, and Georgios Lekkas. Personalized finance advisory through case-based recommender systems and diversification strategies. *Decision Support Systems*, 77:100 – 111, 2015.

[21] Xia Ning, Christian Desrosiers, and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In Francesco Ricci, Lior Rokach, and Bracha Shapira, editors, *Recommender Systems Handbook*, pages 37–76. Springer US, 2015.

[22] J.M. Noguera, M.J. Barranco, R.J. Segura, and L. Martínez. A mobile 3d-gis hybrid recommender system for tourism. *Information Sciences*, 215:37–52, 2012.

[23] Michael P O'Mahony, Neil J Hurley, and Guénolé Silvestre. Detecting noise in recommender system databases. In *Proceedings of the 11th international conference on Intelligent user interfaces*, pages 109–115. ACM, 2006.

[24] Hau Xuan Pham and Jason J Jung. Preference-based user rating correction process for interactive recommendation systems. *Multimedia tools and applications*, 65(1):119–132, 2013.

[25] István Pilászy and Domonkos Tikk. Recommending new movies: even a few ratings are more valuable than metadata. In *Proceedings of the third ACM conference on Recommender systems*, pages 93–100. ACM, 2009.

[26] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In

Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, pages 175–186, New York, USA, 1994. ACM.

[27] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

[28] Raciel Yera Toledo and Yailé Caballero Mota. An e-learning collaborative filtering approach to suggest problems to solve in programming online judges. *International Journal of Distance Education Technologies*, 12(2):51–65, 2014.

[29] Raciel Yera Toledo, Yailé Caballero Mota, and Luis Martínez. Correcting noisy ratings in collaborative recommender systems. *Knowledge-Based Systems*, 76:96 – 108, 2015.

[30] Valentina Zanardi and Licia Capra. Dynamic updating of online recommender systems via feed-forward controllers. In *Proceedings of the 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, pages 11–19. ACM, 2011.

Appendix A. Notation used in CFRS and GRS

To formalize the problem behind CFRSs and GRS we use specific notation:

- $U = \{u_1, \ldots, u_m\}$ is the set of users.
- $I = \{i_1, \ldots, i_n\}$ is the set of items.
- $R \subseteq U \times I$ is the set of known ratings.

- $r_{u_j i_k} \in R$ is the rating given by user u_j over item i_k.
- $R_{u_j \bullet}$ are the ratings given by user u_j.
- $R_{i_k \bullet}$ are the ratings over item i_k.
- $G_a = \{m_1, \ldots, m_r\} \subseteq G$ is the target group. G_a members are noted by aliases, $m_1, \ldots, m_r$. A user may belong to several groups, being G the set of all groups.
- $R_{G_a \bullet}$ are the ratings provided by the members of G_a.
- $R_{G_a i_k}$ are the ratings provided by G_a members over item i_k.
- c_{u_j} and c_{i_k} is the class of u_j and i_k, respectively.
- R^*, $R^*_{u_j,\bullet}$, and $R^*_{G_a,\bullet}$ is the result of the NNM over the corresponding rating set.

Table A.8 clarifies the usage of the notation over the commonly used ratings table. For the sake of clearness, the group members are shown together, but there is no restriction over the users that can form a group

Table A.8: Notation used in the algorithms with and their respective interpretation in terms of the set of elements that they refer to.

I			i_1	$\ldots$	i_k	i_l	$\ldots$	i_n	
U		u_1	$r_{u_1 i_1}$	$\ldots$	$r_{u_1 i_k}$	$r_{u_1 i_l}$	$\ldots$	$r_{u_1 i_n}$	$R_{\bullet,i_k}$
		$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$	$r_{u_j i_k}$
G_a		u_j	$r_{u_j i_1}$	$\ldots$	$r_{u_j i_k}$	$r_{u_i i_l}$	$\ldots$	$r_{u_j i_n}$	$R_{u_j,\bullet}$
		$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$	
	m_1	u_x	$r_{u_x i_1}$	$\ldots$	$r_{u_x i_k}$	$r_{u_x i_l}$	$\ldots$	$r_{u_x i_n}$	$R_{G_a,\bullet}$
	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$	
	m_r	u_z	$r_{u_z i_1}$	$\ldots$	$r_{u_z i_k}$	$r_{u_z i_l}$	$\ldots$	$r_{u_z i_n}$	R_{G_a,i_l}
		$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$	
		u_m	$r_{u_m i_1}$	$\ldots$	$r_{u_m i_k}$	$r_{u_m i_l}$	$\ldots$	$r_{u_m i_n}$	

***Biographical Note**

Luis Martínez (M'10) received the M.Sc. and Ph.D. degrees in Computer Sciences, both from the University of Granada, Spain, in 1993 and 1999, respectively. Currently, he is Full Professor of Computer Science Department and Head of ICT Research Centre at the University of Jaén. His current research interests are linguistic preference modeling, decision making, fuzzy logic based systems, computer aided learning, sensory evaluation, recommender systems and electronic commerce. He co-edited nine journal special issues on fuzzy preference modelling, soft computing, linguistic decision making and fuzzy sets theory and published more than 75 papers in journals indexed by the SCI as well as 30 book chapters and more than 120 contributions in International Conferences related to his areas. In 2015, he authored a monograph *The 2-tuple Linguistic Model: Computing with Words in Decision Making* (Springer).

Dr. Martínez is member of the European Society for Fuzzy Logic and Technology and IEEE, Co-Editor in Chief of the of the International Journal of Computational Intelligence Systems and an Associate Editor of the journals IEEE TRANSACTIONS ON FUZZY SYSTEMS, Information Fusion, the International Journal of Fuzzy Systems, Journal of Intelligent & Fuzzy Systems, the Scientific World Journal, Journal of Fuzzy Mathematics and serves as member of the journal Editorial Board of the Journal of Universal Computer Sciences. He received twice the IEEE TRANSACTIONS ON FUZZY SYSTEMS Outstanding Paper Award 2008 and 2012 (bestowed in 2011 and 2015 respectively). He is also Visiting Professor in University of Technology Sydney, Guest Professor in the Southwest Jiaotong University and honourable professor in Xihua University both in Chengdu (China).