Social Tracks: Recommender System for Multiple Individuals using Social Influence

Lesly Alejandra Gonzalez Camacho[a], João Henrique Kersul Faria,
Solange Nice Alves-Souza[b] and Lucia Vilela Leite Filgueiras[c]
Departamento de Engenharia de Computação e Sistemas Digitais,
Escola Politécnica da Universidade de São Paulo, São Paulo, Brasil

Keywords: Recommender System, Group Recommender System, Social Network, Social Influence.

Abstract: The number of data generated through interactions within a social network, or interactions within a platform resources (eg. clicks, hits, purchases), grow exponentially over time. The popularization of social networks and the increase of interactions allow data to be analyzed to predict the tastes and desires of consumers. The use of recommendation systems to filter content based on the characteristics and tastes of a user is already widespread and applied across platforms. However, the application of recommendation systems to multiple individuals is a less explored field. For this project, data was gathered from social networks to recommend music playlists to a group of individuals. Listening to music as a group is a common activity, be it with friends, couples or in parties. Social network data are used to identify the social influence of the individuals in the group. In addition, to identify the preferences, the characteristics of the songs most frequently heard by the members of the group are assembled. Matrix factorization is used to predict group interests. Proposed influence factor, based on similarity, leadership and expertise, is added to compute a final recommendation. A social network was created to support the controlled experiment, the results show the prediction made by the system vary of 1,455 of the ratings made by the group' members.

1 INTRODUCTION

Current applications increasingly use recommender systems (RS) that suggest products and/or services of interest to their users. Platforms, e-commerce and social networks, such as Netflix, Spotify, Amazon, and Facebook use RS to recommend content more suited to the interests of individuals. The proposal of an RS is to generate personalized recommendations for each user using algorithms that evaluate the items of their interest, based on identified preferences, as well as data of other users with similar interests.

Recommendations based on a user's characteristics and preferences are already widely disseminated and applied across platforms (Contratres et al., 2018). However, activities such as traveling, playing, watching movies, or listening to music, can and are often performed by groups of people. In these situations, an RS should go beyond individual ratings and evaluate the preferences of the group so that the result of the recommendation is satisfactory to all. This approach is known as the Group Recommendation System (GRS) (Ricci et al., 2011) and is one of the aspects little explored in RS. Thus, GRS identifies interests common to individuals in the group to generate recommendations.

As examples of applications for GRS, or with multiple individuals, one has to obtain a better place for traveling in a group, TV that can adapt its programming according to the people who are watching it, songs to be played in a car with several passengers, among others.

Connected phones and the popularization of social networks increase the generation and daily data flow (Quirino et al., 2015). Social networks data is rich in information about how individuals relate, form groups, share common tastes, and influence each other. Several papers have been published in recent years using social networking data to improve product and service recommendations (Zhang et al., 2013; Zhao et al., 2016; Lian et al., 2016; Prando et al., 2017; Contratres et al., 2018; Gonzalez-Camacho and Alves-Souza, 2018).

[a] https://orcid.org/0000-0002-9387-8351
[b] https://orcid.org/0000-0002-6112-3536
[c] https://orcid.org/0000-0003-3791-6269

Camacho, L., Faria, J., Alves-Souza, S. and Filgueiras, L.
Social Tracks: Recommender System for Multiple Individuals using Social Influence.
DOI: 10.5220/0008166503630371
In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2019)*, pages 363-371
ISBN: 978-989-758-382-7

We here propose a RS that, based on the definition of the members of the group, uses data from a social network to identify social influence among these individuals. An experiment, which includes a mobile application that interfaces with the proposed RS, is presented to validate the proposal. This application is used to choose the group of friends and, in the end, to present the result of the recommendation. From the mobile application, the social network data and the individual musical preferences of a musical platform are recovered. Based on the calculated social influence, a subset of the set of songs formed by every-one's preference is recommended to the group.

The social influence is calculated based on four factors and has the greatest weight in identifying the songs recommended to the group.

This work is divided as follows; Section II provides details about the characteristics and techniques employed in RS and GRS, besides discussing the data of social networks and their use in RS. Section III presents the RS proposed, evidencing the calculation proposed to identify the social influence on the group. In Section IV, the RS architecture and details of the experiment are presented. Section V; presents the results and Section VI, the conclusion of the work.

2 RECOMMENDER SYSTEMS

In social relationships, it is natural for people to recommend books, music, and movies to one another. RS has come to assist and to extend this natural process of content exchange (Resnick and Varian, 1997), is generally used when there are many items to choose from and it is unfeasible for the user to be aware of all the content available (Deng et al., 2014; Lalwani et al., 2015; Al-Hassan et al., 2015)

There are different filtering approaches employed in RS, among which the most popular is Collaborative Filtering (CF) (Ricci et al., 2011). CF aims to calculate the similarity of a user in relation to others, to recommend the active user items that other users with similar likes preferred in the past.

CF is divided into two main models: *Neighborhood Methods* and *Latent Factor Models*. Neighborhood methods focus on calculating the relationship between items, or between users, based on the ratings made to the items by the active user, or by their more similar neighbors (Koren et al., 2009). In contrast to neighborhood methods, which use the classifications stored directly in the forecast, Latent Factor models use these classifications to learn a predictive model. The idea is to be able to model user-item interactions with factors that represent latent characteristics of users and items, such as the user preference class and the item category class (Desrosiers and Karypis, 2011).

Latent factor models operate by characterizing items and users in computationally inferred factors based on user ratings. These factors are comparable to the item categories, such as the genre of a movie (Koren et al., 2009).

Matrix Factorization (MF) is a latent factor model approach (Koren et al., 2009). In this approach, each item i that can be recommended is described by a vector q_i of latent factors estimated from the characteristics of the item. For each user u, there is another vector p_u that represents the user's interest in the items. Equation 1: estimates a user's prediction for an item.

$$r_{ui} = q_i \cdot p_u \quad (1)$$

In computing terms, the highest processing cost occurs in the calculation of q_i and p_u and an efficient method to do this is the Alternating Least Squares (ALS) (Koren et al., 2009; Koren and Bell, 2011), which allows calculations to occur in parallel since the parameters of the matrix are calculated independently. In addition, ALS is the most efficient in systems based on implicit data (Hu et al., 2008). In music recommendations, one does not always have explicit data of the user's rating for a particular song, but it is implicitly inferred through the number of times he has heard it.

2.1 Aggregation Techniques

Aggregation techniques are used to combine individual recommendations, generating a recommendation that can satisfy a users group.

Some of the main aggregation techniques are (Masthoff, 2011):

- Average: Gets the average of the rating for a given item.
- Multiplicative: multiplies the rating of each individual, obtaining a value for the rating of the group.
- Vote by approval: counts the number of times an item has been evaluated above a certain threshold.
- Minor dissatisfaction: consider the lowest rating as the group's assessment.
- Greater satisfaction: considers the highest rating for the group rating.
- Average without dissatisfaction: performs the average of the ratings, disregarding items with ratings lower than a threshold value.

Each aggregation strategy generates a group assessment for a particular item or set of items. For example, using the ”Average” technique, if an item is rated by user A with grade 8, by B with 9, and by C with 10, the rating of the group consisting of A, B, and C is 9 among the three ratings).

2.2 Social Influence Features in the Recommendation to Groups

Differently from individual recommendation, in a group recommendation, other social interactions need to be taken into account. For example, the perception of the whole group could be affected in case the recommendation is in any way embarrassing for one of the group members, (Masthoff and Gatt, 2006).

Masthoff and Gatt (2006) explains that the satisfaction of a group can depend on two causes: (i) emotional contagion, in which the satisfaction or dissatisfaction of some users can lead to the satisfaction, or not, of others; (ii) compliance, in which the opinion of others may influence a user’s opinion, whether by normative or information influence. Normative is when the individual, by wanting to be part of a group, expresses an opinion equal to that of the group, even if inwardly he/she does not agree. Conversely, informational is when the opinion of the individual, in fact, changes, because he/she believes that the group is correct.

Wang and Lu (2014) discuss the concept of influence in which a user may have greater power of contagion and compliance in a group. Discovering the user with the greatest influence in a group becomes relevant, because satisfying this user, increases the chances of satisfying the whole group. Zhu and Huberman (2014) show how people’s choices are affected by the recommendations of others.

Masthoff (2011) combines the aggregation techniques with the mechanisms of social psychology, ending in three possible strategies that can only be implemented if RS has a feedback mechanism in which user satisfaction can be measured in real time. These strategies are:

- Strongly support the grumpier: This strategy recommends the item that the least satisfied person likes most.
- Weakly supportive: This strategy selects items that are reasonably appreciated by the least satisfied member (items rated 8/10 or above).
- With weights: This strategy adds weights to the users, depending on their satisfaction, and thus, uses the weights at the moment of performing the aggregation.

3 THE RECOMMENDER SYSTEM PROPOSED

The RS is accessed by the using a mobile application designed to simulate interaction with e-commerce. This application (detailed below) accesses the user’s social network data and the data about the musical preferences of the members of the group on a musical platform. The group of friends and credentials of the user in the social network are informed in the login to access the application. This information is sent to the RS, which is composed of the following modules:

- Social influence calculator: it calculates the influence of each individual in relation to the group based on the proposed influence model.
- Individual recommendation calculator: it calculates the recommendation for each member of the group using data obtained from the musical platform.
- Aggregation techniques: it aggregates the individual recommendations considering the influence. This module uses the aggregation techniques detailed in 2.1 with the result of applying the proposed influence model to create the recommendation for the group.
- Music recommendation: A subset of songs computed as a result of the recommendation.

In the proposed RS the individual recommendations are identified using CF, implementing MF, or more specifically, ALS. The ALS is used to infer user preference for a set of items, using the implicit ratings of other users.

Masthoff (2011) details two experiments conducted to assess which aggregation technique made the most sense for a group, between ”Minor Dissatisfaction”, ”Average” and ”Average without dissatisfaction”. The authors concluded that, although it is simple, the technique of ”Media without dissatisfaction” generates good results. Thus, in this work, the technique ”Media without dissatisfaction” is used to aggregate individual recommendations.

3.1 Social Influence Modeling

Guo et al. (2016) details the influence of an individual on other people, showing how this influence can be a key factor for the recommendation. The authors use five factors as the basis for calculating an user’s influence on a group:

- Expertise Factor: expresses a user’s knowledge of a given topic. Generally, the opinion of experts is more accepted than that of others.

- Susceptibility Factor: considers how much a user is susceptible to the opinions and emotions from others.
- Personality Factor: defined by the individual's behavior pattern. Guo et al. (2016) determines personality factor performing Thomas Kilmann Instrument (TKI), which identifies the behavioral trend of an individual to deal with the other members of the group.
- Intimacy Factor: Measures how much a group is connected. The more the members of a group are close, the more likely they are to accept the opinions of the others.
- Similarity Factor: evaluates the degree of similarity among group members. Obtained through activities and information in common.

In the approach proposed by Guo et al. (2016), influence factors are difficult to automate. For example, to define the *Personality Factor* the TKI test is performed, which consists in making a series of interviews with each member of the group. Albeit interesting, in an actual application, the time to take the test will certainly discourage an individual from continuing into the environment to receive the recommendation. That is why the factors proposed by Guo et al. (2016) are impracticable for a RS by the way these factors are calculated.

Differently from Guo et al. (2016), 3 factors to define the social influence are proposed here to determine the preference of groups of users. These factors are calculated through data coming from social networks only. These factors are the Expert Factor, the Leader Factor, and the Similarity Factor. They are normalized on a scale from 0 to 1 and they are detailed as follows:

- Expert Factor: evaluates a user's knowledge of a specific topic. For music, it measures how much a user hears and knows artists.

$$E_{ij} = \sum_{i=1}^{nu} \sum_{i \neq j, j=1}^{nu} \frac{\frac{nf_i}{nf_i+nf_j} + \frac{na_i}{na_i+na_j} + \frac{nm_i}{nm_i+nm_j}}{3} \quad (2)$$

Where E_{ij} is the element of matrix E that measures the users *expertise*, nu is the number of users in the group, nf is the number of followers in the user's music application, na is the number of artists the user follows, and nm is the number of songs the user has in his/her library.

$$Fexp_i = \frac{\sum_{j=1}^{nu} E_{ij}}{nu} \quad (3)$$

Where $Fexp_i$ is the *Expert Factor* for user i, nu is the number of users in the group.

- Leader Factor: measures how much a user represents the figure of a leader in a group. It is measured by the repercussion of his/her posts in relation to the other members of the group.

$$L_{ij} = \sum_{i=1}^{nu} \sum_{i \neq j, j=1}^{nu} \frac{\frac{tl_i}{tl_i+tl_j} + \frac{tm_i}{tm_i+tm_j}}{2} \quad (4)$$

Where L_{ij} is the element of matrix L for evaluating the repercussion of a users' posts, nu is the number of users in the group, tl_i is the total number of Likes user i received, tm_i is the total number of mentions user i received. Analogous to tl_j and tm_j.

$$Flead_i = \frac{\sum_{j=1}^{nu} L_{ij}}{nu} \quad (5)$$

Where $Flead_i$ is the *Leader Factor* for user i and nu is the number of users in the group.

- Similarity Factor: Measures how similar a user is to the others in a group. This factor is calculated by observing the activities in common among users within the social network.

$$S_{ij} = \sum_{i=1}^{nu} \sum_{i \neq j, j=1}^{nu} \frac{\frac{l_{j,i}+l_{i,j}}{tl_j+tl_i} + \frac{s_{j,i}+s_{i,j}}{ts_j+ts_i}}{2} \quad (6)$$

Where S_{ij} is the element of matrix S that lists the posts of interest and shared by each user. S_{ij} is calculated between each pair of users, nu is the number of users in the group, $l_{j,i}$ is the number of posts user j liked of user i, $s_{j,i}$ is the number of posts that user j shared of user i. Analogous to $l_{i,j}$ and $s_{i,j}$.

$$Fsim_i = \frac{\sum_{j=1}^{nu} S_{ij}}{nu} \quad (7)$$

Where $Fsim_i$ is the *Similarity Factor* for user i and nu is the number of users in the group and S_{ij}.

- Influence Factor: The influence factor is the average of the three previously defined factors and which is used for aggregating of the individual recommendations.

$$Finf_i = (Fexp_i + Flead_i + Fsim_i)/3 \quad (8)$$

Eq. 9 includes the Influence Factor to calculate the individual recommendations.

$$r_{influence_i} = r_{i,j} * (1 + Finf_i) \quad (9)$$

Where $r_{influence_i}$ is the rating of a user i considering his/her influence, $r_{i,j}$ is a user i's rating for a given item j and $Finf_i$ is the influence factor of i. Thus, the influence factor changes the rating calculated by the ALS for each user.

Finally, the rating of an item by the group ($RG_{g,j}$) (eq. 10) is the aggregation of individual ratings considering his/her social influence.

$$RG_{g,j} = \sum_{i=1}^{nu} r_{influence_i} \tag{10}$$

Where nu is the number of users in the group and $r_{influence_i}$ is the rating of a user i considering his/her influence.

4 ARCHITECTURE AND IMPLEMENTATION

Figure 1 shows the proposed RS, called Social Tracks, architecture divided into three main parts:

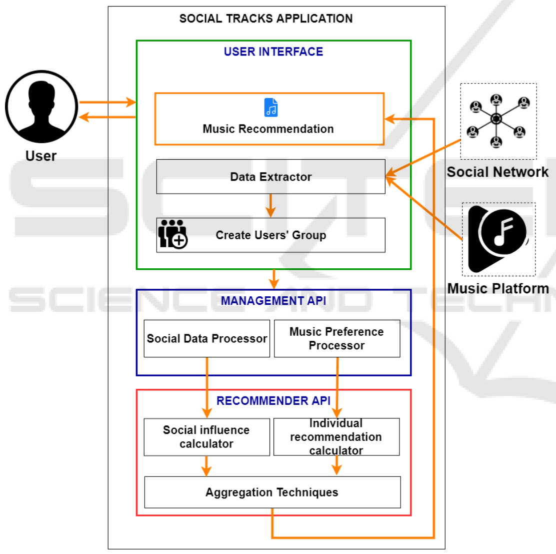

Figure 1: The RS Architecture proposed.

1. Management API: makes the connection between the mobile application and the recommendation API. Here data are organized, formatted and sent to the Recommender API. Later the results are also organized and suitable to be shown in the Mobile Application to the user.
2. Recommender API: processes users' ratings for a set of items as well as social network information to generate a recommendation based on social influence.
3. User Interface: allows the user to interact with RS. Through this, implemented by a mobile application, the user provides both data from his/her music activity (Spotify) and from his/her social network, creates and manages the groups, receives the recommendations for a selected group and can listen to and rate the list of recommended songs.

4.1 Recommendation Data-flow

Figure 2 details the recommendation process and the type of information generated in each step. The flow of recommendation (Figure 1) begins with a user connecting to the application, using his/her credentials from the music platform and later the social network credentials.

Next, the user defines a new group or chooses one of the already existing ones, from which he/she will listen to songs (activities that will be done in a group). The application then retrieves data from the music activities such as the name of singers, or the bands, the name of songs, and the total number of times the user listened to the songs. In addition, the application also retrieves data from the social activity of the group of friends, including the user's own, such as mentions, comments, likes, and information sharing.

The hypothesis is the friends in the group already acquired items in e-commerce. Therefore, the preference data of friends are known in the e-commerce. Thus, in the controlled experiment, the group preferences in the musical platform are also collected.

Using MF, more precisely the ALS, the data of the musical activities are used to generate recommendations of music for each member in the group.

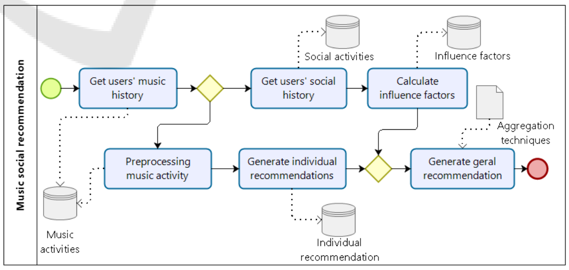

Figure 2: Social Tracks' recommendation process.

Social data is used to calculate Leader and Similarity Factors. The Expert Factor, as opposed to the others, is calculated with the data of the musical activity of each user. Since all these factors have been computed, the influence of each individual in relation to the group is calculated and also the individual recommendation. Thus, the system aggregates the individual recommendations considering the influence, obtaining the recommendation of the set of songs for the initially selected group. At the end of the process, users receive, through the mobile application, the set of recommended songs, which can be heard and rated. The rating makes possible to evaluate whether the result of the recommendation was assertive or not for the group.

The individual recommendations calculated using ALS is the most expensive step in the flow processing (Figure 2) because the set of data is very large since the database has thousands of users and thousands of songs to determine the latent factors.

Although the calculation of the influence factors algorithm (Figure 2) is computationally expensive, it is done in a short time, since the groups are limited to a maximum of 5 individuals. This limitation was imposed by the computational resources employed in the experiment to meet the system requirements making recommendations in an acceptable time.

4.2 Controlled Environment

The controlled experiment use a social network (SocialTracks Mastodon) created to ensure that information exchange occurs in the social network as well as music information in the music platform for the group that will receive the recommendation.

Mastodon (https://mastodon.social) was used to create a social network; it was chosen because it is open source, easy to use and it has features similar to Twitter, which is a widely used social network with fewer restrictions on accessing its users' data.

The instance of Mastodon was created to hold up to 100 users. All the computational limitations were dictated by the existing resources.

In the second stage of the research the experiment will be repeated in an uncontrolled environment to compare the results. The result of the second stage of the research will be timely disclosed in a future paper.

4.3 Data for the RS Training

4.3.1 Music Service Platform Dataset

Last.fm's public API was used to collect the dataset that served as the basis for training the ALS algorithm. Last.fm API is an online aggregator of musical data that allowed the crawling of 105,655 songs, rated by 41,242 users, with a total of 309,986 implicit ratings. In this API, all users' data are publicly available. This dataset was used to simulate a music service platform with a considerable number of users and enough music to generate recommendations through the ALS algorithm. Table 1 details this dataset. The User-Song data in this table is an example of implicit rating because it provides no direct rating of songs, but rather the number of times a user has heard a particular song.

Table 1: Last.fm dataset information details.

Data type	Quantity	Details
Song	105.655	Name, duration and artist (name of singer, band or group)
User	41.262	Name, country, age, and playcount (total number of times the user listened to songs)
User-Song	309.986	User, song and playcount (number of times the user listened to a particular song)

4.3.2 Individual Musical Preferences Dataset

Spotify was the music platform used to obtain individual musical preferences of the experiment' participants (Figures 1 and 2). It was chosen because, besides being a widely known platform, it provides an easy-to-use REST API, allowing an easy integration with web/mobile application. Spotify API returns JSON metadata about artists, albums, and music tracks that the logged-in user listened to in Spotify.

The preferences of each participant are collected when he/she logs into the application. Then, these data are preprocessed and added to Last.fm's dataset to carry out the training. This was necessary because of the ALS algorithm, which only generates the recommendation of new items if the user is part of the training dataset. Part of the preprocessing step was filtering individual musical preferences to contain only the songs matching the set of songs available in the Last.fm dataset.

The algorithm then evaluates the users' recommendations based on the latent factors. A list of the 50 best-rated songs for each user is generated.

5 TEST AND RESULTS

The results presented here constitute a proof of concept for the proposed RS. For this, 11 participants allowed forming 10 groups with a varied number of members.

Table 2: Computation of influence factors.

Group	UserId	F_{exp}	F_{lead}	F_{sim}	F_{inf}
1	2	0.8184	0.2619	0.1666	0.4156
	11	0.6260	0.3611	0.0416	0.3429
	5	0.1909	0.4166	0.0833	0.2303
	8	0.3644	0.2936	0.0416	0.2332
2	9	0.4945	0.4448	0.4771	0.4722
	6	0.4016	0.2758	0.1953	0.2909
	3	0.6037	0.2792	0.5225	0.4685
3	9	0.5660	0.5555	0.5	0.5405
	6	0.4339	0.1111	0.5	0.3483
4	6	0.5046	0.4652	0.0750	0.3483
	11	0.6319	0.1583	0	0.2634
	9	0.5805	0.5069	0.1583	0.4152
	4	0.5176	0	0	0.1725
	8	0.2651	0.4527	0.0833	0.2670
5	2	0.7021	0.4166	0.3571	0.4919
	4	0.3179	0.2444	0.4625	0.3416
	3	0.4798	0.3388	0.3196	0.3794
6	9	0.7668	0.2500	0.25	0.4222
	5	0.2273	0	0	0.0757
	8	0.5058	0.5833	0.25	0.4463
7	2	0.6583	0.3737	0.5	0.5106
	3	0.3416	0.2929	0.5	0.3781
8	6	0.6114	0.5256	0.1369	0.4246
	10	0.0903	0	0.0625	0.0509
	5	0.2196	0.2896	0.0333	0.1809
	7	0.7853	0.3169	0.0750	0.3924
	3	0.7098	0.5343	0.1410	0.4617
9	2	0.9887	0.4166	0.5	0.6351
	10	0.0112	0.2500	0.5	0.2537
10	2	0.7381	0.4470	0.1450	0.4434
	3	0.6078	0.4789	0.1290	0.4052
	10	0.0562	0.3161	0.0357	0.1360
	9	0.5493	0.5258	0.1254	0.4002
	8	0.3573	0.2825	0.0177	0.2192
	1	0.7486	0	0	0.2495
	7	0.6807	0.3371	0.0274	0.3484
	4	0.5178	0.1600	0.0542	0.2440
	6	0.5006	0.4659	0.0997	0.3554
	5	0.2060	0.2825	0.0187	0.1691

Table 2 exhibits the influence factors calculated for each member from each group.

Table 2 shows each group and its members. User data has been masked for privacy reasons, being the users identified by an ID varying from 1 to 11 (field userId).

To generate the recommendation for a group, the following is computed:

- individual recommendations made (Figure 2) using user's data retrieved from Spotify, and
- user's influence factor calculated using data retrieved from a social network.

The similarity factor (F_{sim}) is influenced by the user's interaction in the social network; in other words, the F_{sim} depends on how active a user is in the social network. This factor can be different for two users, even though they are very similar. For example, consider two users i and j that give many likes to posts made by each of them, mutually. However, user i usually shares more posts from user j than the opposite; then F_{sim} for user i will be greater than for user j. Besides, F_{sim} for a user may be zero if he/she only usually posts in the social networks, without sharing, or giving likes, to the posts from the other friends. In the same way, the leader factor ($Flead$) might be zero for an individual that did not receive likes or mentions from the others in the group.

The final tests with an appropriate number of users (Krejcie and Morgan, 1970) will be performed and presented in a future paper.

5.1 Group Recommendation Results

The Social Tracks Application (Figure 1) is also employed to allow users to evaluate the result of recommendations. For this, 20 tracks are returned to each participant in the group that gives a grade from 1 to 5 to each music. The root mean square error (RMSE) (Gonzalez-Camacho and Alves-Souza, 2018) is used to measure the difference between rates given by users and the system. The GRS generates values in the range of [0, 2] (Eq. 10). Therefore, GRS results were normalized to values in the interval of [1, 5] (Eq. 11).

$$R_{System}(g,j) = \left\lfloor \frac{RG_{g,j}}{\max\left(RG_{g,j}\right)} * 4 \right\rfloor + 1 \qquad (11)$$

Where $R_{System}(g,j)$ is the normalization of j's item rating for a given g group. $RG_{g,j}$ is defined in Eq. 10.

Tables 3 and 4 introduce part of the results for 2 groups. All the results from the experiment are available in https://github.com/SocialTracksDataAnalise.

Table 3 shows three tracks recommended to group 1, and Table 4 shows those to the group 5. Field $RG_{g,j}$ illustrates the prediction values given by the system and R_{system} is its respective normalized value. UserId identifies each member in the group; R_u is the track rating made by a respective member and R_{Av} is the average of the ratings for the track, calculated from R_u, for the group.

Table 3: Part of track rating results of group 1.

Group	Track Name	$RG_{g,j}$	R_{system}	UserId	R_u	R_{Av}
1	Sex on Fire	1.62	4	8	5	4
				5	1	
				2	5	
				11	5	
	The Scientist	1.66	4	8	4	4.25
				5	5	
				2	4	
				11	4	
	In the End	1.60	4	8	4	4
				5	5	
				2	3	
				11	4	

Table 4: Part of track rating results of group 5.

Group	Track Name	$RG_{g,j}$	R_{system}	UserId	R_u	R_{Av}
5	Sex on Fire	1.35	3	2	5	4.33
				3	4	
				4	4	
	The Scientist	1.42	4	2	4	3.66
				3	3	
				4	4	
	Mr. Brightside	1.43	4	2	2	3.33
				3	3	
				4	5	

RMSE is given by:

$$RMSE = \sqrt{\frac{1}{K}\sum_{i=1}^{N}\sum_{j=1}^{M_i}(\hat{R}_{i,j} - R_{i,j})^2} \quad (12)$$

$$K = \sum_{i=1}^{N}\sum_{j=1}^{M_i} 1 \quad (13)$$

In eq.12, $\hat{R}_{i,j} = R_{system}(g, j)$, it is the rating that the system made for a track j, which is recommended for group i. $R_{i,j}$ is the average rating that a group i made for a recommended track j. N is the total number of groups participating and M_i is the number of recommendations made for the group i,

The RMSE value varies between 0 and 4 (Contratres et al., 2018), because as $\left|\hat{R}_{i,j} - R_{i,j}\right| \leq 4$, we have $0 \leq (\hat{R}_{i,j} - R_{i,j})^2 \leq 4^2$, then:

$$0 \leq \sqrt{\frac{1}{K}\sum_{i=1}^{N}\sum_{j=1}^{M_i}(\hat{R}_{i,j} - R_{i,j})^2} \leq 4 \quad (14)$$

The RMSE value obtained for this experiment was:

$$RMSE = 1.4554 \quad (15)$$

According the RMSE value, the variation between the ratings given by GRS and members of the groups was 1.5 approximately.

Figure 3 exhibits two distribution of ratings of the pieces of music, one given by the system ($R_{-system}$) and the other, given by the group of users (R_{-av}). These results confirm that the music recommendations made for the groups were well appreciated, once the ratings made by the groups were 1.45 points above the prediction.

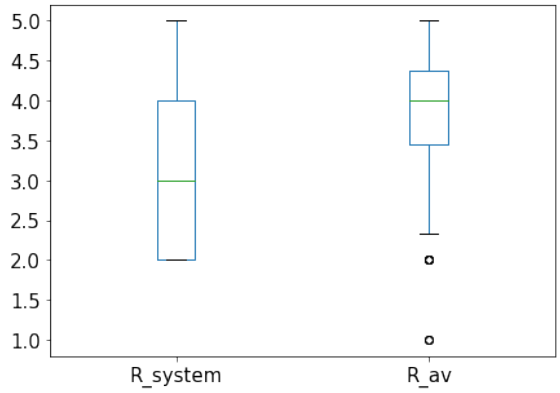

Figure 3: Distribution of prediction given by GRS and by the groups.

6 CONCLUSIONS

The proposed RS makes the recommendation for groups of individuals considering the influence factor among them, which is calculated based on three factors: expert, leader and similarity. In our proposal, these factors are calculated using information from a social network. Controlled experiment allowed testing our proposal, guaranteeing that the members of the groups had some activity in the social network. For a proof of concept, the controlled experiment was conducted with 10 users to demonstrate the effectiveness of our proposal. As a result, the assertiveness of the RS, computed based on RMSE was 68.7%. As future work, we intend to integrate the influence factor within the ALS algorithm for calculating the latent factors. As lessons learned, besides the difficulty in managing users for taking part in the experiment, a user-test based on user-experience should be conducted to help to prepare the user instruction material to facilitate her/his participation. The contributions of this paper are: (i) the influence factor that consider the capacity of leadership, expertise in a subject and the preference similarity among individuals that will do a common activity. These factors are calculated using only information retrieved from social networks; (ii) the use data from social network to improve recommendation. As a future work the influence factor, also with data extracted from social networks, will be proposed to improve the recommendation for a cold-start scenario.

ACKNOWLEDGEMENTS

The authors are grateful for the support given by São Paulo Research Foundation (FAPESP). Grant #2014/04851-8, and the support given by Itaú Unibanco S.A. trough the Itaú Scholarship Program, at the Centro de Ciência de Dados (C^2D), Universidade de São Paulo, Brazil.

REFERENCES

Al-Hassan, M., Lu, H., and Lu, J. (2015). A semantic enhanced hybrid recommendation approach: A case study of e-Government tourism service recommendation system. *Decis. Support Syst.*, 72:97–109.

Contratres, F. G., Alves-Souza, S. N., Filgueira, L. V. L., and DeSouza, L. S. (2018). Sentiment analysis of social network data for cold-start relief in recommender systems. In *New Contributions in Information Systems and Technologies. Advances in Intelligent Systems and Computing*. in press.

Deng, S., Huang, L., and Xu, G. (2014). Social network-based service recommendation with trust enhancement. *Expert Systems with Applications*, 41(18):8075–8084.

Desrosiers, C. and Karypis, G. (2011). *A Comprehensive Survey of Neighborhood-based Recommendation Methods*, chapter 4. Springer US.

Gonzalez-Camacho, L. A. and Alves-Souza, S. N. (2018). Social network data to alleviate cold-start in recommender system: A systematic review. *Information Processing & Management*, 54(4):529–544.

Guo, J., Zhu, Y., Li, A., Wang, Q., and Han, W. (2016). A Social Influence Approach for Group User Modeling in Group Recommendation Systems. *IEEE Intelligent Systems*, 31(5):40–48.

Hu, Y., Volinsky, C., and Koren, Y. (2008). Collaborative filtering for implicit feedback datasets. *Proceedings - IEEE International Conference on Data Mining, ICDM*.

Koren, Y. and Bell, R. (2011). *Advances in Collaborative Filtering*, chapter 5. Springer US.

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Krejcie, R. V. and Morgan, D. W. (1970). Determining Sample Size for Research Activities . *Educational and Psychological Measurement*, 30(3).

Lalwani, D., Somayajulu, D. V. L. N., and Krishna, P. R. (2015). A community driven social recommendation system. *Proc. - 2015 IEEE Int. Conf. Big Data, IEEE Big Data 2015*, pages 821–826.

Lian, D., Ge, Y., Zhang, F., Yuan, N. J., Xie, X., Zhou, T., and Rui, Y. (2016). Content-aware collaborative filtering for location recommendation based on human mobility data. *Proc. - IEEE Int. Conf. Data Mining, ICDM*, 2016-Janua:261–270.

Masthoff, J. (2011). *Group Recommender Systems: Combining Individual Models*, chapter 21. Springer US.

Masthoff, J. and Gatt, A. (2006). In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. *User Modeling and User-Adapted Interaction*, 16(3-4):281–319.

Prando, A., Contratres, F., Alves-Souza, S., and deSouza, L. (2017). Content-based recommender system using social networks for cold-start users. In *Proceedings of the 9th International Joint Conference on Knowledge Discovery,Knowledge Engineering and Knowledge Management*. in press.

Quirino, G. Z., Mals, N. P., Groterhorst, V. M., De Souza, S. N., and De Souza, L. S. (2015). Meneduca - Social school network to support the educational environment. In *Proceedings - 2015 41st Latin American Computing Conference, CLEI 2015*.

Resnick, P. and Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, 40(3):56–58.

Ricci, F., Rokach, L., and Shapira, B. (2011). *Recommender Systems Handbook*. Springer US.

Wang, Z. and Lu, H. (2014). Online recommender system based on social network regularization. 8834:487–494.

Zhang, Y., Chen, W., and Yin, Z. (2013). Collaborative filtering with social regularization for TV program recommendation. *Knowledge-Based Syst.*, 54:310–317.

Zhao, W. X., Li, S., He, Y., Chang, E. Y., Wen, J. R., and Li, X. (2016). Connecting Social Media to E-Commerce: Cold-Start Product Recommendation Using Microblogging Information. *IEEE Trans. Knowl. Data Eng.*, 28:1147–1159.

Zhu, H. and Huberman, B. A. (2014). To Switch or Not To Switch: Understanding Social Influence in Online Choices. *American Behavioral Scientist*, 58(10):1329–1344.