

iVoyage: Interactive Tour Recommendation System

Teresa Jerard

Dept. of Computer Science and Engineering Mar
Athanasius College of Engineering Kothamangalam,
Kerala

Anandu Sabu

Dept. of Computer Science and Engineering Mar
Athanasius College of Engineering Kothamangalam,
Kerala

Aparna Asok

Dept. of Computer Science and Engineering Mar
Athanasius College of Engineering Kothamangalam,
Kerala

Prof. Neethu Subash

Dept. of Computer Science and Engineering Mar
Athanasius College of Engineering Kothamangalam,
Kerala

Abstract—Planning for a vacation that takes into consideration all the travel preferences of an individual can be tedious. An interactive tourism recommendation system is designed to provide travel itineraries to users by applying different machine learning models. This design involves taking into consideration user's preferences regarding attraction spots, cuisine type etc. Restricted Boltzmann Machine model, content-based filtering model, location-based context-aware model and predictive analysis are used in this application. The project also utilizes crowd information to support tour planning by displaying a bar graph of relative crowdedness in a day and letting users change their itinerary accordingly in order to produce a revised itinerary plan. In the proposed model we use sentiment analysis to aggregate the preferences and opinions of each individual of the tour group to provide an itinerary plan for a group.

Keywords—Tour recommendation system, Restricted Boltzmann Machine, Content-based filtering, Predictive analysis, Location-based context-aware system, Sentiment analysis, Crowd information.

I. INTRODUCTION

It can be time-consuming to plan a holiday that considers all of an individual's travel preferences and interests. Even when a set of attraction sites or hotels are found people do not have individual information regarding the places. Especially in terms of crowd information or crowd density. And when group traveling is required, an accurate itinerary must include the interests and preferences of each individual of the group. Using various machine learning models, an immersive tourism recommendation framework is designed to provide users with travel itineraries. This architecture takes into account the user's tastes in terms of tourist attractions, cuisine, and hotel amenities. The design includes the Restricted Boltzmann Machine (RBM) model, content-based filtering model, location-based context-aware model, and predictive analysis.

Tourists can prepare their tour plans and itineraries using recommender systems. When group traveling is required an accurate itinerary must include the interests and preferences of each individual of the group. To provide the group with the most appropriate itinerary, sentiment analysis is used. Each individual may have their own preferences according to which they receive personal itineraries. To provide a group itinerary,

every user is allowed to review and rate every other users' recommendations. When sentiment analysis is performed on these reviews, we can find the places with most positive reviews and those places are considered to form the new itinerary.

II. RELATED WORKS

In general, attraction recommendation uses collaborative-filter (CF) based and content-based methods. Users' traveling histories and user-location relationships are used for collaborative-filter-based methods. In [1] with help of a combination of Markov model and topic model, attractions are predicted using the history of traveler. Based on collaborative filtering, [2] excavated knowledge from GPS data in order to discover locations, and activities and a shared matrix factorization is employed for recommendation. In case of content-based methods, it utilizes geo-tagged photos in social networking websites. In [4] based on the geo-location of people large amount of geo-tagged photos were gathered. By using text or images user can find similar tourist spots.

Karanikolaou [6] conducted a study to understand human crowd behavior that was developed in 'human-centered participatory sensing'. The traffic of people in tourist spots is being studied based on the viewpoint of route allocation of tourist buses, keeping the fact of traffic congestion in tour buses [7, 8]. Uchida [9] generated an itinerary using simulation based optimization.

In case of restaurant recommendation, traditional recommendation systems use collaborative filtering (CF) to recommend products to a particular user. This type of filtering uses the idea that similar users vote for similar items. Memory-based collaborative filtering or user-based collaborative filtering algorithms try to find a correlation among users based on voting pattern of different users. This correlation is used to get predictions for similar users. This results in big computation and memory loads on the system if the prediction is done in real time. Hence, these systems are not popular in real time applications [12]. Instead of using only user preferences some systems use social views of a city as well. Model-based CF obtain recommendations use a complex model of

the check in data from Location-based Social networking. In Ji Bao's [13] work he generated using Hyperlink-Induced Topic Search algorithm. This work was further extended by Xin Cao [14] in which the association between the locations and user information is considered but these works do not take in into account the feature of different location categories.

Content-based filtering is the most common approach used in hotel recommendation systems. The properties of the items rated by previous users and the best-matching results are recommended. In this context, the local popularity of the best hotels is based on ratings given by the users, which is used as the main feature for the content based filtering approach [11]. The main shortcomings of this approach are that it is limited by the number and the types of features associated with the objects for recommendations, it may involve the issue of over-specialization as no inherent method is included for finding something unexpected and it may not predict precisely for new users. Usually, a content-based recommendation system needs enough ratings to provide accurate recommendations.

In general, group recommendations can be generated either by aggregating individual recommendations or user profiles [17]. In the first case, a recommendation is generated for each group member, and these recommendations are merged into one group recommendation. In the second case, all group member's preferences are aggregated to a virtual user, for whom a recommendation is made. Different strategies to aggregate user preferences exist, many inspired by Social Choice Theory [18]. Another method proposed was using public displays [15] which uses a shared display of recommendations which allows users to share their opinions. Nevertheless, public displays raise privacy issues. Some people do not feel comfortable entering data into a public device [16].

III. PROPOSED SYSTEM OVERVIEW

The overall architecture of the system is shown in Fig. 1. Datasets for attraction recommendation, hotel recommendation and restaurant recommendation are collected and is under-gone through data transformation and preprocessing. These processed datasets along with user preferences are then given as input to Restricted Boltzmann Machine model, location-based context-aware model and predictive analysis. Also, a revised itinerary is formed by providing users' with the option to change their itinerary based on crowd information. Fig. 2 shows the architecture for generating the group itinerary. Each group member is provided with an individual itinerary based on their preferences. These itineraries are then reviewed by the other users and sentiment analysis is performed on these reviews to obtain the final itinerary.

IV. PROPOSED METHODOLOGY

A. Dataset

In the case of the attraction recommendation model TripAdvisor dataset was collected. The dataset included the attributes province, category, attraction id, city, country, name, price, rating, average pricing, latitude, and longitude for each city province. The address of the attraction was taken with the

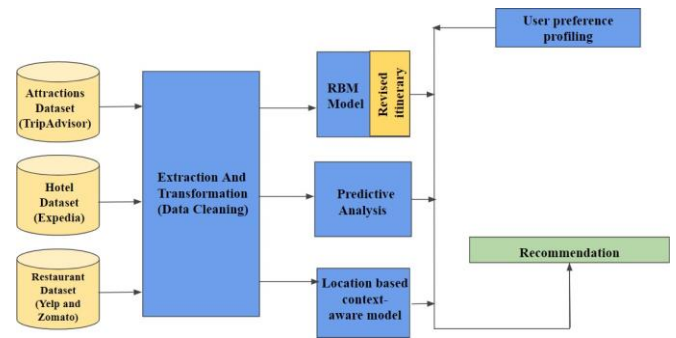


Fig. 1. Pipeline of various models

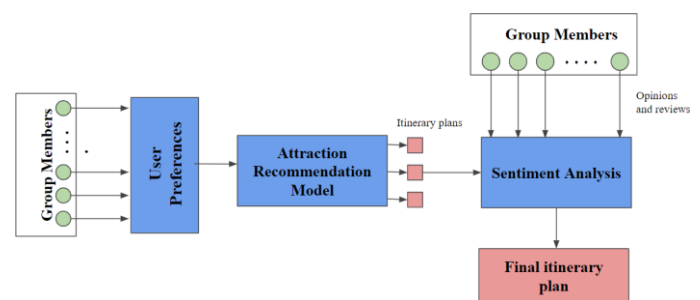


Fig. 2. Pipeline of formation of group itinerary

help of geocoding, a Google API used to obtain the coordinates of each attraction spots. The location coordinates of an attraction were assumed to be the average location of attraction categories in a city. Unavailable prices and ratings were also averaged based on city and type of attraction.

We used Google's 'popular times' as the dataset for crowd information filtering. Google obtains this information based on the data that is combined from users with enabled locations on their devices. 'Popular times' gives information in the form of a graph with respect to 24 hours of a day for each of the places [16]. Averaging of several weeks of data is also done, which includes data on large crowds during events and festivals, where the staying time is longer.

For the restaurant recommendation model, Zomato dataset is used to obtain restaurant information. Another dataset is used regarding user details to include the user location. The Zomato dataset includes the attributes name, rate, address, location, restaurant type, cuisine type, approximate price, reviews list, phone number, and city. The user dataset includes user id and their latitude and longitude.

For the hotel recommendation model, Expedia's dataset was used. This dataset gives details of the user interactions with their booking websites like their location, destinations searched, hotels booked, etc. This dataset was published by Expedia and most of the user ids and locations were hidden or scrambled to protect their privacy.

The attributes that were present in the dataset are the date and time stamp, site name, site continent, user's location, the distance between the hotel and the user at the time of searching, user-id unique for each user, hotel logs including the start and end date of stay, booking status which specifies whether the search for the hotel leads to booking a reservation or not, number of adults and children, destination id unique for the hotels of each destination user searched for, the hotel's country and hotel market in that destination.

B. Data Preprocessing

For the purpose of training an RBM model each user's information regarding a set of tourist spots rating is to be used. Consequently from the dataset information of the individual user and their corresponding set of attraction spots with respective rating values are to be used while the rest are discarded. The attractions dataset was used to create the user rating matrix. This matrix consists of r rows for users and c columns for each attraction present in the dataset. The ratings provided by the user i for the attraction j are given at the cell $[i, j]$ of the matrix. The position without any user rating is given the value as 0 ratings. In the restaurant recommendation model, we need to do data cleaning and data preprocessing to proceed further. Data cleaning includes the following steps like deleting unnecessary columns, removing the duplicates, removing the NaN values from the dataset, changing and also, we need to perform certain text preprocessing steps. Before applying content-based filtering to the dataset we need to vectorize the data. That is, in this application, we are converting the data in string format into vectors to perform cosine similarity. In order to vectorize these data, we are using Term Frequency - Inverse document frequency vectorization (TF-IDF). After this step, we get a matrix of each word and its corresponding significance with respect to each restaurant in a numerical format. The formula for TF-IDF vectorization is as follows:

$$\text{tf idf}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D) \quad (1)$$

$$\text{tf}(t, d) = \log(1 + \text{freq}(t, d)) \quad (2)$$

$$\text{idf}(t, D) = \log\left(\frac{N}{\text{count}(d \in D: t \in d)}\right) \quad (3)$$

where, tf is the term frequency, idf is the inverse document frequency, t is the word which is vectorized, d is the document and D is the document set.

To make our model more reliable, we first delete all the hotel searches which did not lead to booking a reservation. This would shorten the available data but it will become more appropriate to our recommendation system. Then we sort the remaining hotel clusters based on three main criteria which are location-based. The first criteria is hotels based on the user's location at the time of search and the distance towards the hotel they searched for. The hotel searched in the destination of the user's choice is the second criterion. The final criteria are the best hotels in the searched destination's country. Then we use the test set as input and predict the five best hotel clusters

from the sorted group in the order of the criteria mentioned above.

V. MODULE-WISE IMPLEMENTATION

A. Attraction Recommendation Module

RBM is a two-layered artificial neural network namely, visible and hidden layer. RBMs have application in regression, feature learning, dimensionality reduction, classification, topic modelling, and collaborative filtering. Due to the absence of connection within the visible layer and the hidden layer, it is called 'restricted'. The two layers are connected with a completely bipartite graph. This means that every node of visible layer will be connected to every node of hidden layer but in a restricted fashion [5]. The difference between the reconstructions and the original input will be high due to the reason that weights of the RBM are randomly assigned. The input to the model is taken as the user's rating for each attraction spot based on the required number of sites for the new user. During the forward pass, each rating is multiplied with weights, and bias is added to it, which is then given as input to the sigmoid activation function.

$$p(h_i = 1 | v) = \frac{1}{1 + e^{-(b_j + w_{ji}v_i)}} \quad (4)$$

Equation (4) gives the equation for the forward pass. Now the output obtained in the hidden layer will act as the new input for the calculation of the visible layer which forms the backward pass (5).

$$p(v_i = 1 | h) = \frac{1}{1 + e^{-(a_i + w_{ih}h_i)}} \quad (5)$$

The two passes form Gibbs sampling. The forward pass and backward pass are represented in fig 3 [2]. Now the passes are continued until the change in weight obtained is very insignificant. This is the process of contrastive divergence and is given by (6).

$$\Delta W = v_0 \otimes P(h_0 | v_0) - v_k \otimes P(h_k | v_k) \quad (6)$$

In (4) and (5), h_i is the i^{th} hidden layer, v_i is the i^{th} visible layer, b is bias and w represents weights. Now the hidden layer will have the rating value of each site to be of a particular category. This marks the point of getting similar users. Now the similar user's top spots will be recommended to the new user [9].

B. Crowd Information Filtering

In this system, we provide crowd information to change the itinerary based on users' choices. Crowd information is generated using `livepopulartimes` package in python language. This package returns data that contain several details on a place given its address. From this data, we filter out the crowd information. This data contains crowd data in a numerical format for 24 hours every day in a week.

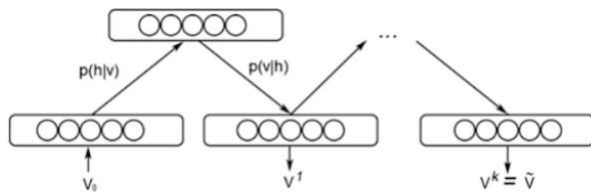


Fig. 3. Forward pass and backward pass process of RBM

This bar graph is displayed for each attraction spot in the itinerary and the users are given the option of whether or not to visit this place. If the user chooses not to visit a place, a revised itinerary plan is recommended with new places.

C. Hotel Recommendation Module

We use previous customer search data to create a model to which we then apply the new data to get the best predictions. First, we go through all the data in the train set and sort out all the search results in which no booking has been done. This will help us find hotels that were confirmed to be booked and shows us the hotel had some form of activity eliminating newer hotels without much experience or engagement. The best set of hotels is predicted based on its location of the user and the destination.

Here we use three main criteria for this purpose, the first parameter is to group all the hotel clusters for which the user's location and destination distance are available. This would give the best result by convenience based on the user's travel route. The next criteria would be to group the hotel clusters based on the destination, this is the next best criteria as it provides all the hotels within the reach of the destination for efficient traveling. The final criteria are to group the hotel clusters based on the destination country, this would be the least preferred criteria but applying these criteria would be treated as the model we get from the historical data of the predictive analysis we use. The user input is then retrieved and their details are applied to the model we just created. We search the available details in order of the above three criteria. If both the user location and destination distance are present in the model then we use the hotel clusters as the suggestions or if it is not enough, we go for the next criteria and search by destination and suggest the satisfying hotel cluster and we consider the third criteria last and suggest the corresponding hotel clusters. We suggest the five best hotels which satisfy anyone of the above criteria. The order of applying the criteria is important as the probability of the best hotels decreases as we move from the first criteria to the last.

D. Restaurant Recommendation Module

A content-based filtering model is a machine learning model that uses a user profile and item profile to compare the similarity between the two and to recommend the items.

In this application, the item corresponds to the restaurant. Hence item profile includes vectored reviews of restaurants that define the restaurant characteristics. User profiles can either be determined through user preferences or users' previous choices. In this case, we are taking a restaurant that the user visited and recommend similar restaurants. To do this we find the cosine similarity measure between the chosen restaurant and every other restaurant. It compares two restaurants on a normalized scale. Cosine similarity is defined as the cosine of the angle between the two vectors which is done by finding the dot product between the two vectors. Let the two vectors be A and B, then cosine similarity is given as follows:

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} \quad (7)$$

$$A \cdot B = \sum_{i=1}^n A_i B_i \quad (8)$$

$$\|A\| \|B\| = \sqrt{\sum_{i=1}^n A_i^2 \sum_{i=1}^n B_i^2} \quad (9)$$

If the angle between two vectors is small, they have high similarity and as the angle increases, the similarity between the items reduces. The similarity measure lies between 0 and 1. If the similarity is closer to 1, then it is more similar and such vectors would have more priority. These cosine similarity measures are stored in the matrix and the restaurants are sorted based on this. The restaurants with high similarity are filtered out.

The location-based context-aware system uses the location of the user as a context to recommend the nearest restaurants. To do this, our application is using Haversine's formula as given in (10) and (11). This formula finds the distance between two points in a spherical structure. Here the earth is considered as a sphere.

$$d = 2 * r * \sin^{-1}(P) \quad (10)$$

$$P = \sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \left(\cos(\phi_2) \cos(\phi_1) \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)\right)} \quad (11)$$

where, r is the radius of sphere, ϕ_1 is latitude of first place, ϕ_2 is latitude of second place, λ_1 is longitude of first place and λ_2 is longitude of second place.

E. Group Itinerary Generation

Sentiment analysis is used for the purpose of sensing positive or negative sentiment in text. In order to generate a group itinerary, we need to consider the opinions and suggestions of each individual of the group. When each itinerary is generated of members, every individual can review the places where they can state their opinions. These opinions are analyzed and rated to be positive, negative, or neutral. The places with the most positive reviews are prioritized and a sorted itinerary will be generated for the complete group. TextBlob is a python library that makes use of Natural Language Toolkit (NLTK)

for Natural Language Processing (NLP). This library maintains complex analysis and procedures on textual data. The semantic orientation and the intensity of each word in a sentence define a sentiment for lexicon-based approaches. For this pre-defined dictionary classifying negative and positive words is required. The final sentiment of a sentence is calculated by taking an average of all the sentiments once we assign individual scores to all words. For a sentence, TextBlob also returns polarity and subjectivity. The polarity of 1 means the given text is a positive sentence, and -1 for a negative sentence. Subjectivity ranges within [0,1]. Subjectivity gives the amount of personal opinion and factual data is contained in the text. When the text contains more personal opinion than factual information, subjectivity is higher. Another parameter TextBlob uses is intensity. Intensity defines whether a word has the power to modify the next word.

VI. RESULTS

The restricted Boltzmann machine provides similar users to that of a newly registered user. The existing similar user's set of attraction sites are filtered such that the difference between each consecutive site must be less by finding the difference between the latitude, longitude set of two places using Haversine's formula. These attraction sites are given as output along with the address, the corresponding rating as well. Error based on increasing epochs is given in Fig. 5. The option of visiting or not visiting based on the crowd information provided to users on each attraction site is used to prioritize the schedule and thereby providing a revised itinerary.

Content-based filtering and location-based context-aware systems can provide users with recommendations that are based on user preferences and past choices' and it also considers the location of the user. Haversine's formula is used to find the location which has an accuracy of 0.996. The comparison between actual distance and calculated distance is shown in Fig. 4. This model takes into consideration restaurant reviews, ratings, cuisine type, and other attributes to find accurate options. Most scalable and faster methods are used here.

User behavior can be highly unpredictable and can vary drastically from one user to the other so instead of ratings, we use location-based filtering and reservation status as the parameters for the recommendation system.

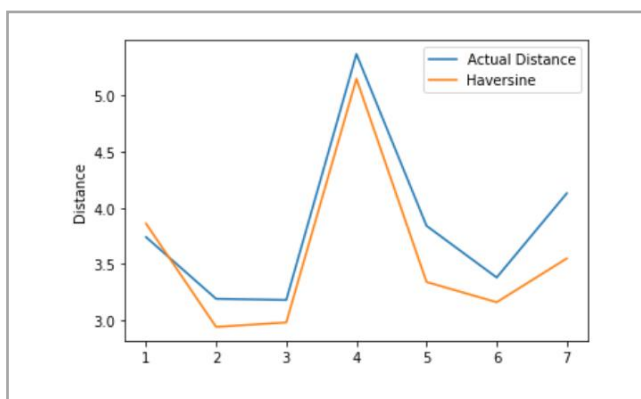


Fig. 4. Actual distance versus distance calculated using Haversine's formula

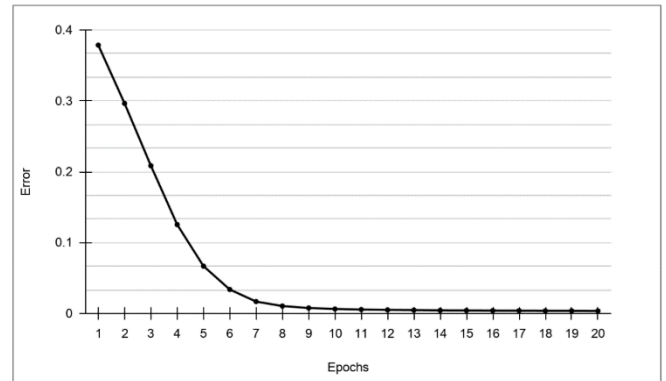


Fig. 5. Error based on increasing epochs

Hotel location is optimized based on the user's location and the destination they searched for and the usage of booking status helps to identify the hotels that have more interaction and experience while removing newer hotels with lesser engagement and experience.

VII. CONCLUSION

Group itineraries are devised using sentiment analysis. It takes into consideration all the recommendations, preferences, reviews, ratings of all users. Each user can enter their reviews on all the attraction spots recommended for every other user based on which new recommendations are built. Based on user selection on the purpose of the tour, budget, dates of travel, and destination, itineraries are devised for attraction spots along with restaurant recommendations and hotel recommendations. Crowd information is provided to form a recomputed itinerary. The sentiment analysis model is used to compute itineraries for group tours taking into account individual opinions.

REFERENCES

- [1] Attraction recommendation- Towards personalized tourism via collective intelligence
- [2] wiki.pathmind.com/restricted-boltzmann-machine
- [3] Modeling Prediction in Recommender Systems Using Restricted Boltzmann Machine Hanene Ben Yedder and Umme Zakia, Aly Ahmed
- [4] <https://www.presentslide.in/2019/08/sentiment-analysis-textblob-library.html>
- [5] <https://www.edureka.co/blog/restricted-boltzmann-machine-tutorial/>
- [6] Karanikolaou, S., Boutsis, I., Kalogeraki, V.: Understanding event attendance through analysis of human crowd behavior in social networks. In: Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems, pp. 322–325. ACM (2014)
- [7] Zhang, L., Wang, Y.P., Sun, J., Yu, B.: The sightseeing bus schedule optimization under park and ride systems in tourist attractions. *Ann Oper Res* 1–19 (2016). <https://doi.org/10.1007/s10479-016-2364-4>
- [8] Hasuike T, Katagiri H, Tsubaki H, Tsuda H (2013) Tour planning for sightseeing with timedependent satisfactions of activities and traveling times. *Am J Oper Res* 3(3):369–379
- [9] Kuriyama H, Murata Y, Shibata N, Yasumoto K (2010) Simultaneous multi-user scheduled cyclic scheduling method considering congestion situation in cities and tourist spots. *Inf Process Soc Jpn Trans Inf Process Soc Jpn* 51(3):885–898
- [10] Popp M (2012) Positive and negative urban tourist crowding: Florence, Italy. *Tour Geogr* 14 (1):50–72
- [11] Saga R, Hayashi Y, Tsuji H. Hotel recommender system based on user's preference transition[C]//Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on. IEEE, 2008: 2437-2442.
- [12] Orozov, T., Narasimhan, N., Vasudevan, V.: Using location for personalized POI recommendations in mobile environments. In: SAINT 2006, pp. 124–129 (2006)

- [13] Bao, J., Zheng, Y., Mokbel, M. F. (2012). Location-based and preference-aware recommendation using sparse geo-social networking data. In Proceedings of the 20th International Conference on Advances in Geographic Information Systems (pp. 199–208).: ACM.
- [14] Cao, X., Cong, G., Jensen, C. S. (2010). Mining significant semantic locations from gps data. Proceedings of the VLDB Endowment, 3(1-2), 1009–1020.
- [15] Daniel Herzog , Wolfgang Wörndl.(2019). A User Study on Groups Interacting with Tourist Trip Recommender Systems in Public Spaces. Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization.
- [16] Harry Brignull and Yvonne Rogers. 2003. Enticing People to Interact with Large Public Displays in Public Spaces. In Human-Computer Interaction INTERACT '03: IFIP TC13 International Conference on Human-Computer Interaction, 1st-5th September 2003, Zurich, Switzerland.
- [17] Judith Masthoff. 2004. Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers. User Modeling and User-Adapted Interaction 14, 1 (Feb. 2004), 37–85.
- [18] Judith Masthoff. 2015. Group Recommender Systems: Aggregation, Satisfaction and Group Attributes. In Recommender Systems Handbook, Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer US, Boston, MA, 743–776.