

A Personalized Recommendation System for NetEase Dating Site

Chaoyue Dai, Feng Qian, Wei Jiang, Zhoutian Wang, Zenghong Wu
Hangzhou R&D Center, NetEase Inc.
No. 599 Wangshang Rd, Hangzhou, P.R. China

{daizhaoyue,hzqianfeng,hzjiangwei,hzwangzhoutian,hzwuzh}@corp.netease.com

ABSTRACT

With the popularity of internet, more and more people try to find friends or dating partners on online dating web sites. Recommending appropriate partners from a large amount of candidates becomes an interesting and challenging problem in the field of recommendation system. Various types of recommendation techniques (e.g., content based recommendation, collaborative filtering and association rule mining) have been proposed to tackle this problem. However most of them ignore the personalization concerns that they (1) mainly consider the hot users or frequent items, (2) cover only a portion of users especially ignoring the long tails, (3) and cannot deal with the cold start problem properly. In this paper, we present a regression based hybrid recommendation system that makes use of matching degree, fancy degree, activity, sincerity, popularity and enthusiasm, to recommend appropriate partners. The experimental evaluation of our recommendation system on a real dating web site shows our strategy is more effective and efficient than its previous version which follows the principle of giving higher priority to the recent active users.

1. INTRODUCTION

With the popularity of internet, more and more people try to find friends or dating partners on online dating web sites. Recommending appropriate partners from a large amount of candidates becomes an interesting and challenging problem in the field of recommendation system [22, 17, 9, 18, 19, 14].

Various forms of recommendation (e.g., content based recommendation, collaborative filtering and association rule mining) have been proposed to tackle this problem. However most of them ignore the personalization concerns that they (1) mainly consider the hot users or frequent items, (2) cover only a portion of users especially ignoring the long tails, (3) and cannot deal with the cold start problem properly.

In this paper, we present a regression based hybrid recommendation system that makes use of the matching degree, fancy degree, activity, sincerity, popularity and enthusiasm,

to recommend appropriate partners. The experimental evaluation of our recommendation shows our strategy is more effective and efficient than its previous version which gives higher priority to the recent active users.

Our current recommendation system (1) firstly extracts the basic user attributes from a large amount of data by adopting a distributed processing platform; (2) secondly builds the user preference models by adopting machine learning methods (i.e., decision tree and logistic regression); (3) and finally recommends appropriate dating partners to users based on personalized preference models.

The advantages of our recommendation system include (1) the considerations of both user preference and mutual matching degrees, (2) and the balance of user satisfaction and system needs (e.g., giving high priorities to fresh users).

The remaining sections are organized as follows. We first review relevant research on recommendation systems in Section 2. We then present our regression based hybrid system for dating recommendation in Section 3, together with the experimental evaluations in Section 4 before concluding our work in Section 5.

2. RELATED WORK

Nowadays there are mainly two kinds of conventional methods for recommendation systems, one is content based filtering[16] and the other is collaborative filtering[8, 12]. Content based filtering focuses on the content information that the users are interested in. Based on the feature analysis and feature extraction of this information, the user preferred items are recommended. However, this method is difficult to apply in “user-to-user” recommendation systems due to the lack of user information and explicit user interests. Collaborative filtering is a popular recommendation whose idea is to discover similar users and recommend their commonly interested items by generating a comparable weighted recommendation value[7]. D. Agarose *et al.* and Y. Koren adopted latent factor matrix decomposition to collaborative filtering[1, 2, 10].

The online “user-to-user” recommendation systems differ from the ordinary “user-to-item” recommendation systems. A matching recommendation is successful only if both of users are satisfied with each other[24, 11]. There have been a few studies on the recommendation of romantic dates for online dating users[13, 5].

The majority of online dating recommendation systems adopt graph-based collaborative filtering algorithms. For example, Cai *et al.* use collaborative filtering algorithms based on the similarity of users’ interests and behaviors[5]. Kitty *et*

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/3.0/>. Obtain permission prior to any use beyond those covered by the license. Contact copyright holder by emailing info@vlldb.org. Articles from this volume were invited to present their results at the 40th International Conference on Very Large Data Bases, September 1st - 5th 2014, Hangzhou, China. *Proceedings of the VLDB Endowment*, Vol. 7, No. 13. Copyright 2014 VLDB Endowment 2150-8097/14/08.

al. proposed a graph mining technique to calculate the similarity of the user preferences according to users' stated preferences and the "user-attribute bipartite network"[11]. In this research a graph-based collaborative filter is adopted as a recommendation system, and the recommended matchings must be clustered artificially by recommending the same set of candidates to the similar subjects. Halo *et al.*[23] consider the reciprocal interactions as an important factor in recommendation. And a collaborative filtering based hybrid algorithm was shown to have good performance in recommending for both one-way and mutual contacts.

When facing the cold start problem that the system has no priori knowledge about the newly coming users or items, these methods tend to recommend hot items or users while the long tail has less opportunity to be recommended. This may result in low recommendation coverage of users. As the number of users increases, these models are hard to update in time and not sensitive to new users. This results in a rising of churn rate of new users due to poor interactive experience. Since the recommendation resources are limited under a certain circumstance, our designed recommendation system also takes personalization into consideration.

3. OVERVIEW OF THE RECOMMENDATION SYSTEM

3.1 System Architecture

Figure 1 illustrates the overall architecture of our system which consists of data analysis module and recommendation module.

First, all the log information generated from the web site system is collected in real-time and stored in our distributed database storage (e.g., MySQL). Then the data analysis module rapidly updates the user and system attributes and their status from the huge log information by using the distributed computing platforms. It then updates the users' features that will be used in the next stage. After gathering the newest user features and system status, the recommendation module implements a series of machine learning methods to recommend a list of dating partners to each user. In this experiment, we adopt epiC[6], an extensible and scalable system as the integrated storage/computing platform(our recommendation system is also available on other big data platforms). It is designed to handle multi-structured data under a united interface by decoupling the concurrent programming model and data processing model, which processes the data-intensive computing automatically in parallel efficiently. In Figure 1, epiC:ES2 denotes Elastic Storage System, which has been designed for write-intensive applications, providing a hybrid storage mechanism to support both OLTP and OLAP. And epiC:VBS denotes Virtual Block Service, a file system federation, specially designed for supporting different file systems by leveraging both performance and utilization, which also supports file systems including legacy systems (POSIX), HDFS/GFS (read-only), Key-value stores (append-only) and mission critical storage (memory-resident).

3.2 Data Analysis Module

The frontend of our web site system will generate log information to record the user behaviors, such as login, viewing, sending message, searching and request for a recommendation list. Our data analysis module processes these data in

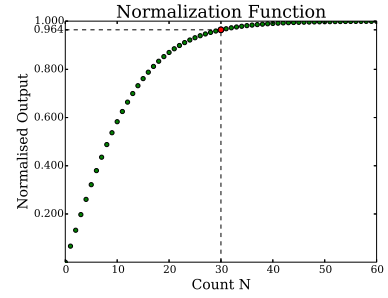


Figure 2: Normalization Function

real-time, so as to update user features and system properties in memory or database. There are many global system properties (e.g., the online number of users and the number of newly registered users) and many other complex indicators. Basic user features associated with each individual user include many dimensions, such as sex, region, age, income, education, ethnicity, occupation, religion, registration time and online time. Besides, they also include some advanced indicators of users to support the needs of our personalized recommendation system. In the following, we describe some important indicators.

Activity: a weighted value indicating the activity of a user that combines various dimensions such as user logins, trends, messages, comments, praises and some other behaviors initiated by user. We denote these behaviors as B and the day d activity of a user A_d is calculated as follows.

$$A_d = \sum_{b \in B} W(n_{b,d}, w_b, s_b). \quad (1)$$

Here $n_{b,d}$ is the action number of behavior b initiated by a user at day d . And w_b and s_b represent the weight and scale parameter of behavior b . And $W(\cdot)$ means a weighted function, defined as below:

$$W(n, w, s) = w \cdot N(n, s), \quad (2)$$

$$N(n, s) = \frac{2}{1 + e^{-4n/s}} - 1. \quad (3)$$

Here $N(n, s)$ is a normalization function which maps a positive number to $[0, 1)$ with a declining slop when n increases. When n is equal to the scale parameter s , $N(n, s)$ equals 0.964. We illustrate the curve of $N(n, s)$ in Figure 2 when s equals to 30. After obtaining one day activity of a user, we summarize a recent 30 day activity of a user by a weighted sum of the previous day's, denoted as A_{30} , equation below:

$$A_{30} = \sum_{d=1}^{30} A_d \cdot e^{-(d\sqrt{\ln 2}/s)^2} / 30. \quad (4)$$

Here s is a attenuation parameter, which we set to 45 in our experiment. Due to the weighted and attenuated sum processing, this indicator can measure a user's true activity comprehensively and effectively, and also handle fraudulence.

Security: a composite index integrated of authenticity of user information, as well as daily behavior of usage of

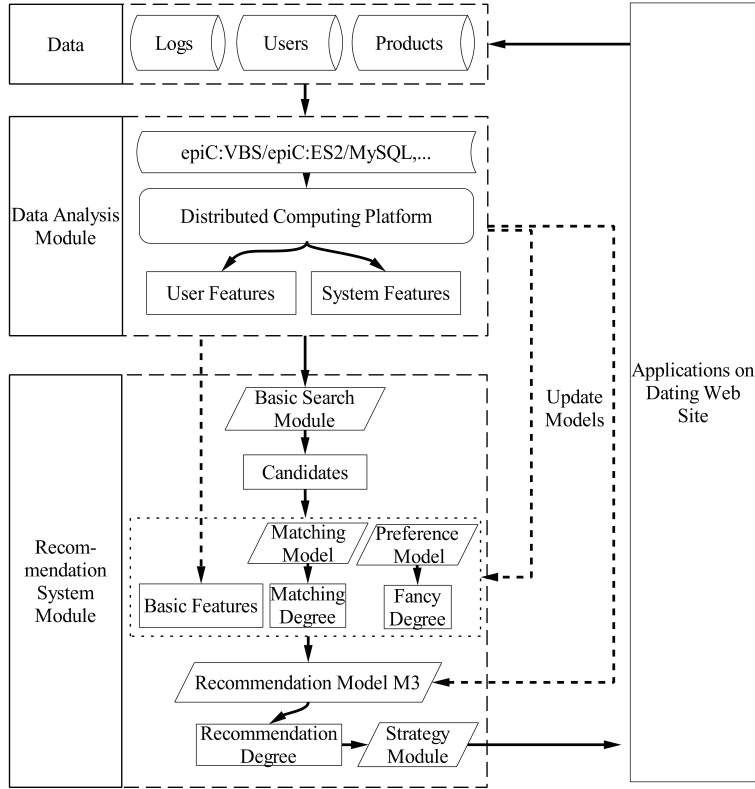


Figure 1: Overall Architecture of our Recommendation System

the application. Authenticity of user information is described by the consistency and reliability of user's information, e.g., whether the head portrait uploaded by user is a real people picture, and whether the user's filled income is consistent with his occupation, position and education. And daily behavior of usage of the application includes whether the user often logins, and how many people who block or report for harassment to him. Finally we obtain a sincerity value range in $[0, 1)$ by the calculation method which is similar to the formula of activity mentioned above. A high sincerity value means the high probability of being a normal user. If the sincerity is very low, the user is very likely a malicious user or rarely participates in the dating web site. By considering the feature of sincerity in recommendation system, we can effectively inhibit the spread of low quality users or malicious users, and let the sincere user be recommended to more sincere people.

Popularity: an index value combined of the receiving number of praises, comments, views, and messages, indicating the popularity of a user in the application. It consists of the recent popularity and the overall popularity. We also calculate it by the weighted and attenuated sum method as used in generating user's activity, but with different parameters and input attributes. Besides, we can even find the fraudulent users or robots with high activity (e.g., frequently send harassment messages) but with very low popularity.

Enthusiasm: an index indicating a user's response to stranger's initiative (e.g., say hi, comment your post, or sending message) based on the statistics of historical communications and feedbacks of a user. Below is a simple formula to calculate enthusiasm Ent .

$$Ent = \frac{m}{N}. \quad (5)$$

where N denotes the different number of people who have taken the initiative to the user and m denotes the number of people this user responses to among the N users. But there is a problem that Ent with low N may not be consistent to the real one. In this case, it will have very high standard deviation and very low confidence level. So we use the lower bound of Wilson score interval[21] as Ent which can solve the problem gracefully, with the formula given below:

$$Ent = \frac{p + \frac{1}{2n}z_{1-\alpha/2}^2 - z_{1-\alpha/2}\sqrt{\frac{p(1-p)}{n} + \frac{z_{1-\alpha/2}^2}{4n^2}}}{1 + \frac{1}{n}z_{1-\alpha/2}^2}. \quad (6)$$

Here p is the observed fraction of positive ratings, which is equal to m/N , n is the total number of ratings, which is the same as N , and $z_{1-\alpha/2}$ is the $1 - \alpha/2$ quantile of the standard normal distribution. Enthusiasm means the probability of the user's active responding to the one who takes the initiative. By considering the enthusiasm of users in the recommendation system, we can improve the exchange activity of the isolated users or not popular users, also suppress the message

harassment of popular users in certain degree, and finally improve the overall user satisfaction.

3.3 Recommendation Module

The recommendation module processes as follows. First, the candidate partners are chosen by basic search module according to the user preference. Then the features of candidates are extracted including many basic features, and some more advanced ones, such as the matching degree, fancy degree, etc. After gathering these features, the recommendation score of each candidates is calculated by user rating prediction model, and finally the recommendation is implemented according to the recommendation score which is adjusted by the strategy module (e.g., giving higher priorities to fresh users).

The basic search module, implemented by Lucene in our system, builds candidates by indexing and searching the basic data with specific search conditions on features. First we index all the features of users in database, and then update indexes increasingly when the users change their information or newly registered users come. When any feature value is modified, the update log service will be notified and record the change. The index updating service updates the index database according to the update log periodically.

The recommendation system implemented in this paper consists of three parts: user matching model, user preference model and user rating prediction model.

3.3.1 User Matching Model

A series of basic dimensions such as age, income and education can be used for matching users. In order to reduce the error rate of man-made rules, we use machine learning methods to obtain a matching model. The training process is carried out as follows.

The training samples can be obtained by some basic features such as age, income, education, nationality, profession and house condition. And the label of training samples can be obtained from the feedbacks such as likes and dislikes from each couple of users. Then we use the decision tree to obtain a matching model, denoted as M1. We adopt C4.5[15] decision tree and GINI Coefficient[20] in our training processes. The *GINI* Coefficient of node t is defined as:

$$GINI(t) = 1 - \sum_j p^2 \left(\frac{j}{t} \right). \quad (7)$$

where $p \left(\frac{j}{t} \right)$ is the ratio of the sample number of class j in node t that denotes the impurity of node t . Its range is between 0 to $1 - 1/k$, where k is the category numbers. We adopt cost complexity pruning algorithm (CART pruning algorithm[4]). The gain of error rate of each split node (non-leaf node), denoted as α , can be measured as

$$\alpha = 1 - \frac{R(t) - R(T_t)}{|N_{T_t}| - 1}. \quad (8)$$

where $|N_{T_t}|$ is the number of sub nodes of node t , $R(t)$ is the error rate cost of node t . It can be represented as follows if node t is pruned.

$$R(t) = r(t) - p(t). \quad (9)$$

where $p(t)$ is the number ratio of node t . $R(T_t)$ denotes the error cost of all sub trees of node t , denoted as T_t , if node t would not be pruned. In this case, $R(T_t)$ is the

sum of all error rates of sub nodes T_t . The advantages of this pruning algorithm include (1) avoiding the over-fitting of training data; (2) lower level of decision tree with higher accuracy and support rate; (3) insensitivity of noisy information. The matching model is effective in case of cold-start recommendation environment for the newly registered users.

3.3.2 User Preference Model

The above user matching model is capable of handling basic matching problem. However it does not take the user preference into account. Therefore we train a user preference model for every user as the representation of his/her personal dating preference, denoted as M2. To a certain extent, fancy degree of a user shows what sort of people he/she likes. For example, if a user has many like-clicking to the users who are teachers, then the user preference model will give a high score to teachers for this user.

The training sample is based on historical dating behavior of individual users (likes or dislikes to other users). The predicted value of model is the user reaction. There are three kinds of user reaction: (1) positive reaction such as sending message, like-clicking and following; (2) negative reaction such as no replying to message and blacklisting; (3) no explicit reaction (e.g., no clicking). The predicted value of positive reaction will get higher recommendation score and vice versa. In this paper, we use decision tree as our prediction model to get a personalized fancy degree to unfamiliar users. After training, the decision tree model is serialized and saved in our distributed storage system. By updating training samples and prediction model in time, our system can utilize the latest user behavior data and provide more precise recommendation.

There are many users with different preferences on the online dating web site. The user preference model can satisfy the dynamic and latent requirements of different users. This model makes the recommendation system satisfy both explicit and implicit personal preferences of users accurately.

3.3.3 User Rating Prediction Model

We introduce a set of dimensions to enhance the recommendation precision. These dimensions include matching degree, fancy degree, activity, sincerity, popularity and enthusiasm. Besides, we also consider whether the users are newly registered or frequent visitors. Based on the above dimensions we train a user rating prediction model for online recommendation.

In our system, the user rating prediction model is established by adopting logistic regression method [3]. The trained model will predict the recommendation rates for candidate users.

In logistic regression method, we denote our feature vector as X_1, X_2, \dots, X_n where n is the number of dimensions. The logistic regression form of recommendation probabilities of candidate users is as follows.

$$p = \frac{1}{1 + \exp -(\beta_0 + \sum_1^n \beta_i X_i)}. \quad (10)$$

Here, $\beta_0, \beta_1, \dots, \beta_n$ are regression coefficients of logistic regression model. In our system we use stochastic gradient descent method [3] to evaluate these regression coefficients. The coefficients are updated online based on the distributed computing system. The trained model is then used to predict recommendation probabilities.

Table 1: Comparison of Coverage

Coverage	Raise percent of V2 to V1
All user	7%
New user	45%
Active user	3%

Table 2: Comparison of Satisfaction

Indicator	Raise percent of V2 to V1
Click ratio	53%
Deep click ratio	177%
User activity	33%

4. EXPERIMENTAL STUDY

In this section, we conduct the experiment on Huatian, a dating web site of NetEase Inc. The data set of the experiment includes user’s information, recommendation data, user’s click data and user’s other interaction data such as say-hi, messages etc.

In this experiment we define the previous recommendation system in Huatian as Recommendation V1 in which the recent active users take precedence over others to be recommended. Recommendation V1 just takes the user active time into account and doesn’t consider users’ interaction history. The recommendation model we introduced in this paper is defined as Recommendation V2. The experiment compares the two strategies in two aspects, user coverage and satisfaction.

4.1 User Coverage

Coverage is a basic performance index in recommendation systems. In our experiment, we investigate it from the following three coverage aspects:

1. All user coverage. To ensure that all registered users could be recommended, a good recommendation model should improve the coverage for all users. We denote C_{all} as all user coverage, U_{rec} as the recommended users and U_{all} as the all registered users. we compute the all user coverage by the formula:

$$C_{all} = U_{rec}/U_{all}.$$

2. New user coverage. A good recommendation system should make efforts to prevent losing newly registered users. In this aspect, the new user coverage index should be measured. We consider the users registered in recent three days as new users. We denote C_{new} as the new user coverage, U_{recNew} as the recommended new users and U_{allNew} as the all new users. The formula is:

$$C_{new} = U_{recNew}/U_{allNew}.$$

3. Active user coverage. In order to increase the user retention, the recommendation system should ensure that the active users are mostly recommended. We denote C_{act} as the active user coverage, U_{recAct} as the recommended active users and U_{allAct} as the all active users. The formula is:

$$C_{act} = U_{recAct}/U_{allAct}.$$

The experiment was based on Huatian recommendation data in recent 7 days, and measures the average value of the results. The experiment list in Table 1.

The experimental results indicate that the recommendation model is better than the previous strategy on the basis of three coverage indexes.

4.2 User Satisfaction

The fundamental purpose of a recommendation system is to improve the user’s satisfaction. In general, user satisfaction can be measured by users click ratio, user’s activity and other interaction. In this experiment, we investigate user’s satisfaction in three aspects including click ratio, deep click ratio and activity:

1. Click ratio. The click ratio is the percentage of the clicks in the recommended users. This index reflects whether the recommended results is compatible for users. We denote CT as the click ratio, U_{click} as the clicked users and U_{rec} as the recommended users. The formula is:

$$CT = U_{click}/U_{rec}.$$

2. Deep click ratio. Some interactions such as messages and followings, are more important than clicks. We call these interactions as deep clicks. The deep click ratio is the percentage of deep clicks in the recommended results. We denote CT_{deep} as the deep click ratio, $U_{deepClick}$ as the deep clicks and U_{rec} as the recommended users. We define deep click ratio as follows.

$$CT_{deep} = U_{deepClick}/U_{rec}.$$

3. User activity. The activity is a weighted value of several factors including clicks, messages, say-hi. This index indicates whether the recommendation system can attract users. We denote AD as user activity, HN as say-hi numbers, CN as the clicks number, MN as message number and U_{all} as all the registered users. User activity is computed as the formula:

$$AD = (HN + CN + MN * 2)/U_{all}.$$

The measurement was also based on Huatian recommendation data in recent 7 days, and we take the average as the results. The experimental results see Table 2.

The results show that the new recommendation model is much effective than the previous model of Huatian with respect to the three satisfaction indicators. In Table 2, the user activity of new recommendation is 33 percent higher than that of the original one, the click ratio is 53 percent higher than that of the original one, the deep click ratio raises 127 percent according to that of the original’s. These results indicate that the recommendation model can satisfy user’s requirements more effectively.

5. CONCLUSIONS

In this paper, we have presented a personalized recommendation system for the NetEase’s Huatian dating web site. Our proposed system (1) extracts the basic user attributes from a large amount of data by adopting a distributed processing platform such as epiC; (2) builds the

user preference models by adopting machine learning methods (i.e., decision tree and logistic regression); (3) and finally recommends appropriate dating partners to users based on preference models and global strategies.

The advantages of our recommendation system include (1) considerations of user preference, matching rate, user popularity and user activity (2) and integrated planning of user needs and product needs.

The experimental evaluation of our personalized dating recommendation system shows our models and strategy are effective and efficient.

6. REFERENCES

- [1] D. Agarwal and B.-C. Chen. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 19–28, 2009.
- [2] D. Agarwal and B.-C. Chen. flda: matrix factorization through latent dirichlet allocation. In *Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010, New York, NY, USA, February 4-6, 2010*, pages 91–100, 2010.
- [3] C. M. Bishop, editor. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [4] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.
- [5] X. Cai, M. Bain, A. Krzywicki, W. Wobcke, Y. S. Kim, P. Compton, and A. Mahidadia. Collaborative filtering for people to people recommendation in social networks. In *Australasian Conference on Artificial Intelligence*, pages 476–485, 2010.
- [6] Dawei Jiang, Gang Chen, Beng Chin Ooi, KianLee Tan, Sai Wu. epic: an extensible and scalable system for processing big data (accepted). *PVLDB*, 7(7), 2014.
- [7] D. M. Fleder and K. Hosanagar. Recommender systems and their impact on sales diversity. In *Proceedings 8th ACM Conference on Electronic Commerce (EC-2007), San Diego, California, USA, June 11-15, 2007*, pages 192–199, 2007.
- [8] J. L. Herlocker, J. A. Konstan, and J. Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, 5(4):287–310, 2002.
- [9] N. Ifada. A tag-based personalized item recommendation system using tensor modeling and topic model approaches. In *Proceedings of The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast, QLD, Australia - July 06 - 11, 2014*, page 1280, 2014.
- [10] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pages 426–434, 2008.
- [11] S. Kutty, R. Nayak, and L. Chen. A people-to-people matching system using graph mining techniques. *World Wide Web*, 17(3):311–349, 2014.
- [12] M. R. McLaughlin and J. L. Herlocker. A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *SIGIR*, pages 329–336, 2004.
- [13] L. A. S. Pizzato, T. Rej, J. Akehurst, I. Koprinska, K. Yacef, and J. Kay. Recommending people to people: the nature of reciprocal recommenders with a case study in online dating. *User Modeling and User-Adapted Interaction*, 23(5):447–488, 2013.
- [14] S. Purushotham and Y. Liu. Collaborative topic regression with social matrix factorization for recommendation systems. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*, 2012.
- [15] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
- [16] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.
- [17] H. Su, K. Zheng, J. Huang, H. Jeung, L. Chen, and X. Zhou. Crowdplanner: A crowd-based route recommendation system. In *Proceedings of IEEE 30th International Conference on Data Engineering, Chicago, ICDE 2014, IL, USA, March 31 - April 4, 2014*, pages 1144–1155, 2014.
- [18] M. Vartak and S. Madden. Chic: a combination-based recommendation system. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22-27, 2013*, pages 981–984, 2013.
- [19] C. Wan, B. Kao, and D. W. Cheung. Location-sensitive resources recommendation in social tagging systems. In *Proceedings of 21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 1960–1964, 2012.
- [20] J. Weymark. Generalized gini indices of equality of opportunity. *The Journal of Economic Inequality*, 1(1):5–24, 2003.
- [21] E. B. Wilson. Probable Inference, the Law of Succession, and Statistical Inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.
- [22] N. B. Yahia, N. B. B. Saoud, and H. B. Ghézala. Community-based collaboration recommendation to support mixed decision-making support. *Journal of Decision Systems*, 23(3):350–371, 2014.
- [23] K. Zhao, X. Wang, M. Yu, and B. Gao. User recommendations in reciprocal and bipartite social networks—an online dating case study. *IEEE Intelligent Systems*, 29(2):27–35, 2014.
- [24] K. A. Zweig and M. Kaufmann. A systematic approach to the one-mode projection of bipartite graphs. *Social Netw. Analys. Mining*, 1(3):187–218, 2011.