IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 10, 2017, accepted January 30, 2018, date of publication February 12, 2018, date of current version March 15, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2801810

A Novel Airbnb Matching Scheme in Shared Economy Using Confidence and Prediction Uncertainty Analysis

LONGHUA GUO[1], JIANHUA LI[1], JIE WU[2], (Fellow, IEEE), WEI CHANG[3], AND JUN WU[1], (Member, IEEE)

[1]Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
[2]Center for Networked Computing, Temple University, Philadelphia, PA 19122, USA
[3]Department of Computer Science, Saint Joseph's University, Philadelphia, PA 19131, USA

Corresponding author: Jun Wu (junwuhn@sjtu.edu.cn)

This work was supported by the National Science Foundation of China under Grant 61571300 and Grant 61431008.

ABSTRACT As a shared economy platform, Airbnb allows customers to collaborate and guides them to hosts' rooms. Based on the records and ratings, it attaches great significance to infer users' satisfaction with their rooms. Several essential problems arise when evaluating satisfaction and matching. Data confidence and prediction bias influence the inference performance of the satisfaction. When two users stay in one room, their joint satisfaction also deserves particular research because of the roommate effect. In this paper, a matching model is built based on the inferred satisfaction considering confidence and prediction uncertainties. The satisfaction with the confidence uncertainty is modeled using a normalized variance of the Beta distribution. The algorithms for inferring satisfaction with the prediction uncertainties are divided into two parts: a weighted matrix factorization-based algorithm for individuals and a preference similarity-based algorithm for pairs. Two matching algorithms are proposed with constraints. Finally, extensive experiments using real-world data show the effectiveness and accuracy of the proposed method.

INDEX TERMS Shared economy, matching scheme, confidence uncertainty, prediction uncertainty.

I. INTRODUCTION

Increasingly popular shared economy [1], [2] and peer-to-peer online platforms [3], such as Airbnb, provide collaborative practices in sharing goods and services like space, skills, and time [4]. Airbnb hosts provide customers with many choices of rooms. When customers have exceedingly positive experiences with the rooms they book in Airbnb, they are likely to use the service again in the future. Airbnb is willing to provide services that recommend customers to hosts to achieve better matching. Matching customers with appropriate rooms is important to Airbnb because it improves customer satisfaction and increases loyalty. However, the customers' satisfaction with rooms are unknown before check-in. Different types of rooms are offered with different limitations for accommodation [5]. As a result, matching customers with rooms while considering uncertainties is challenging.

Fig. 1 illustrates this problem: there are three rooms with different accommodation constrictions. Rooms 1 and 2 are able to accommodate at most two people, while Room 3 is suitable for a single person. With various preferences, customers show a different satisfaction with the provided rooms. The satisfaction indicates the probability for the customer

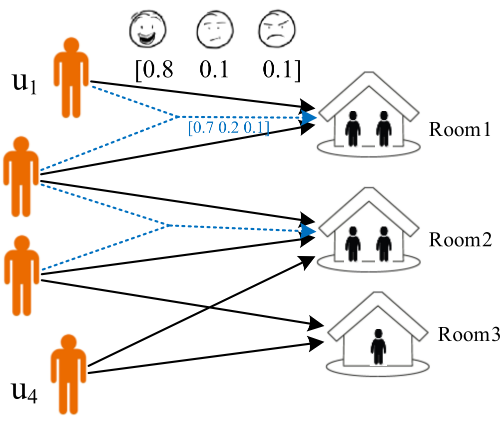


FIGURE 1. Example of the matching problem in Airbnb.

living in the room, which can be modeled using the probabilities of being {happiness, uncertainty, and unhappiness}. The probability of happiness minus that of unhappiness is called utility. For example in Fig. 1, the satisfaction of u_1 staying in Room 1 is {0.8, 0.1, 0.1}. The joint satisfaction of u_1 and u_2 staying together in Room 1 is {0.7, 0.2, 0.1}. The problem seeks to assign customers to rooms while maximizing the total satisfaction and considering the uncertainties. Given the history of the check-in and rating records, modeling the satisfaction with uncertainties is another important step before the matching process. Uncertainties are caused by data confidence and prediction bias. If u_1 had stayed in Room 1 before, the uncertainty, caused by data confidence, is called ***confidence uncertainty***. The more nights the customer lives in a room, the more the customer knows and the higher the confidence of the rating will be. Otherwise, the uncertainty caused by the prediction bias using the inference algorithms, is called ***prediction uncertainty***.

Some customers treat the condition of the provided room in a strict way even if there is only small bias. As a result, we need to match rooms to these customers with low uncertainty. In this paper, we explore the matching problem in the shared economy of Airbnb, and we propose a satisfaction-based matching scheme under the uncertainties. The limitations of the existing works are as follows. First, the confidence of the rating values in recommendation systems is not well considered. For example, customers who stay in a room for a longer time know more about an Airbnb room. Their ratings should be assigned with higher confidence. Second, Airbnb ranks matching results based on the preference features through learning algorithm. Most recommendations works including Airbnb aim to minimize the overall error between prediction value and ground truth. However, the prediction bias and uncertainty for each prediction are seldom modeled in previous works. Third, the joint satisfaction of pairs is not simply the average of satisfactions of two individuals. The bipartite matching problem with conflicts needs research. The scheme seeks to maximize the total satisfaction.

To overcome these limitations, we build an effective and more realistic model that infers users' satisfaction with their rooms. Our work provides algorithms for matchings considering the room and customer constraints, and it demonstrates the effectiveness of the proposed method. To achieve our goal, we face three main challenges. First, it is difficult to model and measure the confidence and prediction uncertainties. Calculating the satisfaction under the confidence and prediction uncertainties is also challenging. Second, the rooms are able to accommodate either one or two users. The records of two users living in a room are much more sparse than that of one user in a room. Considering that having roommates can effect satisfaction, users' joint satisfaction must be specially considered. Third, there are some constraints in matching. For example, each customer can occupy only one room. The number of people in a room is constrained. Balancing the utility and uncertainty in the matching algorithm is challenging.

This paper systematically models the matching problem in Airbnb. The confidence uncertainty and the prediction uncertainty are taken into consideration when inferring the satisfactions. The satisfaction with the confidence uncertainty is calculated based on the number of nights stayed. The inference of the satisfaction with the prediction uncertainty is divided into two parts: individual satisfaction for one user and pair satisfaction for two users. Individual satisfaction utilizes the matrix factorization algorithm to predict the unknown ratings and evaluates the uncertainty based on its category. The pair-joint satisfaction with the prediction uncertainty is modeled based on the community. Afterwards, the satisfaction is evaluated using a unified metric, *Score*. The objective is to maximize the summation of *Score* with the constrains. A matching algorithms is proposed with the satisfaction and constraints. Our main contributions are summarized as follows.

- Utilizing check-in records, the satisfaction between users and rooms is calculated with uncertainty. The problem can eventually be reduced to the matching problem to balance utility and uncertainties. The proposed mechanism takes data confidence and prediction bias into consideration, which achieves more accurate matching.
- The basic matrix factorization minimizes the overall errors of predictions. In our work, the trust level is utilized to calculate the uncertainty of each prediction result, which establishes a basis for matching optimization.
- In the state-of-the-art work, the influence of roommate is not considered. The joint satisfaction of pairs is not simply the average of satisfactions of two individuals. In our model, the community is detected and the joint satisfaction is inferred based on pairs with known ratings in the same community.
- We prove that this user-room matching problem is NP-hard. Two greedy algorithms for maximum matching and maximal matching are proposed. The maximum matching maximizes the summation of matching scores. The maximal matching optimizes the matching relationships while ensuring that each user has a room. Extensive experiments show the effectiveness of our solutions.

The remainder of the paper is organized as follows. Section II describes the related work. Section III presents the satisfaction and matching model and formulates the problem. In Section IV and V, the details of the proposed method are given, including the algorithms for individual satisfaction, pair satisfaction, and the matching problem. Experiments are conducted using real-world data to evaluate the proposed method in Section VI. Section VII further discusses the proposed method. Finally, we draw our conclusion and gave the future work in Section VIII.

II. RELATED WORKS

Airbnb provides guests and hosts recommendation services for better matching results. Given input query information like location, price, and dates, Airbnb ranks high listings whose features are appealing to guests [6]. To evaluate the matching degree, preference features (location relevance, reviews, host response time) and past booking history are utilized for calculating similarities by learning algorithms. With the development of recommendation techniques, matrix factorization (MF) has proven to be one of the most powerful methods of collaborative filtering [7]. To represent the relationships between users and items, latent factors are utilized to model their correlation [8] as $\langle U_i, V_j \rangle$. With these two low-rank latent vectors, the rating matrix can be approximated to fill in missing ratings. To learn the factor vectors, batch gradient descent (BGD) or stochastic gradient descent (SGD) algorithms are utilized to minimize the error. [9] builds a App-permission bipartite graph to estimate the secure score based on the similarity of Apps and permissions. [10] models users decision making process and addresses data sparsity and user/location cold-start problem. In the check-in records of Airbnb, the number of nights stayed are important when considering the certainty of users' rating. However, previous work doesn't consider this influence when filling in missing ratings. The previous works try to minimize the overall error of all the prediction, but the uncertainty of each predicted result is never modeled.

In previous works, there are methods that deal with matching problems under uncertainty. Stochastic programming methods [11] are widely utilized in risk-neutral applications. In the matching dominance method for uncertain objects, [12] captures the semantics of dominance for multi-dimensional uncertain objects. However, these methods can not directly be applied in the proposed problem, because they do not focus on the data confidence. Robust optimization methods consider the worst-case outcome in risk-averse applications. [13] proposes two spatial matching algorithms to minimize the maximum matching distance for applications considering the worst-cases. Unfortunately, uncertainties are not well defined and modeled in previous works.

Creating matching relationships with satisfactions including happiness, uncertainty, and unhappiness metrics is vital. Different metrics are measured and utilized in evaluations and in making decisions in previous works. These methods can be categorized as threshold-based and combination models. In threshold-based models, threshold of one metric is utilized for filtering in the first step and further choice is made on the basis of the remaining metric. For example, [14] proposes a risk threshold-based model to determine allow or deny decisions. In combination models, scores of metrics are integrated to make decision. For example, [15] calculates the combination score using a weighted trust score and a matching score to select the agent. [9], [16] utilizes portfolio theory in the field of finance to balance risks and rewards. Reference [17] proposes a multi-objective optimization strategy with the evolutionary algorithm. Reference [18] proposed a greedy algorithm with an approximation guarantee to solve the weighted bipartite B-matching problem. However, these methods cannot be directly applied to the problem proposed in this paper due to the matching constraints. For example, the joint score of u_1 and u_2 if matched with Room 1 is the summation of the scores when u_1 and u_2 are matched with Room 1 individually in [18]. While the joint score are independent with the individual scores in our scenario. As a result, we need to propose new graph computing method to solve the matching problem.

III. MODEL AND PROBLEM FORMULATION

Here, we first present the uncertainty in the satisfaction between customers and rooms. Afterwards, the bipartite graph model for matching is introduced. Essential symbols and their descriptions are given as shown in Table 1.

TABLE 1. Essential symbols and descriptions.

Symbols	Description
SC	satisfaction with the confidence uncertainty
SP	satisfaction with the prediction uncertainty
L	the number of the nights user had stayed for
Q	the quantity of people who can occupy a room, constrained to 1 or 2
P_h	the probability of happiness stayed in the room
P_u	the probability of uncertainty stayed in the room
P_d	the probability of unhappiness stayed in the room
P_h	the gain from each customer with l_j
P_u	the probability of uncertainty stayed in the room
P_d	the probability of unhappiness stayed in the room
P_h	the gain from each customer with l_j

A. SATISFACTION MODEL WITH UNCERTAINTY

Given a set of users U and rooms R, we assume that only single rooms and double rooms are provided in the platform. One user who stays in a room is called an **Individual**, while two users are called a **Pair**. Satisfaction is a tuple of preference between users and rooms, represented by the happiness probability P_h, uncertainty probability P_u, and unhappiness probability P_d.

Definition 1 (Individual-Room-Rating (IRR) Matrix): The cell in $IRR[i][j]$ records the i_{th} individual's rating of the j_{th} room.

Definition 2 (Pair-Room-Rating (PRR) Matrix): The cell in $PRS[i, j][k]$ records the pair's rating of the k_{th} room. The pair is made up of i_{th} and j_{th} individual

Definition 3 (Individual-Room-Satisfaction (IRS) Matrix): The cell in $IRS[i][j]$ records the i_{th} individual's satisfaction of the j_{th} room.

Definition 4 (Pair-Room-Satisfaction (PRS) Matrix): The cell in $PRS[i,j][k]$ records the pair's satisfaction in the k_{th} room.

Definition 5 (Individual-Category-Satisfaction (ICS) Matrix): The cell in $ICS[i][j]$ records the i_{th} individual satisfaction of the j_{th} category.

In the historical records, ratings represent users' room preferences. We assume that if the rating is more than a threshold, the user likes the room and has a happy experience. Otherwise, the users are unhappy when living in the room. For example, 3-stars is set as the threshold to distinguish happiness and unhappiness for the 0-5 star rating system in this paper. Category is a room classification based on attributes such as location, price, property type, etc.

To distinguish the cause of the uncertainty, the satisfaction is divided into two types: satisfaction with the confidence uncertainty, denoted as SC, and satisfaction with the prediction uncertainty, denoted as SP. The number of the nights stayed is denoted as L. The larger L is, the more confident the rating is. This means that there is a lower uncertainty probability in SC. If the rate is high, the user will be more likely to enjoy the room and will have more certainty of staying in the room next time. If $IRR[i][j]$ exists in the records, SC_{ij} is calculated directly using the records. Otherwise, SP_{ij} is inferred using the existing records.

B. MATCHING MODEL WITH UNCERTAINTY

Assigning rooms is essentially a match between customers and rooms. In this paper, the customer can occupy only one room. The quantity of people who can occupy a room, represented as Q, is constrained to 1 or 2. With a higher P_h and a lower P_d, the user will have a higher probability of liking the room. Taking uncertainty into consideration, Score (calculated by the weighted harmonic mean) is utilized to measure satisfaction, according to the F-measure metric in evaluating the recommendation system [19]. Since confidence and prediction uncertainties effect in the matching model differently, we assign them different weights in the satisfaction score calculation.

$$Score = \frac{(1+\theta)(P_h - P_d)\theta(1-P_u)}{|P_h - P_d| + \theta(1-P_u)} \tag{1}$$

where $\theta = [\theta_1, \theta_2]$ is the weight of uncertainty and $[\theta_1, \theta_2]$ represents the weight of the confidence and prediction uncertainties, respectively. $0 < \theta_1, \theta_2 \leq 1$. The absolute value of the score increases with the decrease of uncertainty.

Using users' satisfaction *Score* of rooms, the objective of the matching problem can be modeled as a weighted Bipartite Matching subject to user and room constraints. Given $G = (U \cup R, E, Score, Q)$, the objective is to find the subgraph $H = (U \cup R, E')$ maximising $\sum_{e \in E'} Score(e)$ with every vertex $u \in U$ incident to at most one edge. U and R are both the nodes in the graph.

$$u \in U : |\{e = (u, r_i) \in E' : r_i \in R\}| \leq 1 \tag{2}$$

and vertex $u \in U$ adjacent to degree constraints

$$r \in R : |\{e = (u_i, r) \in E' : u_i \in U\}| \leq Q(r) \tag{3}$$

where $Q(r) \in \{1, 2\}$.

C. FRAMEWORK

Fig. 2 illustrates the framework of the proposed scheme. First of all, user satisfaction with rooms, including SC and SP, is calculated according to the existing records. In the Bayesian analysis, the beta distribution can be used to describe the initial knowledge concerning probability. SC is calculated with a beta distribution in which the ratings and nights stayed in check-in records are regraded as evidence. For SP, two different algorithms, based on categories and communities, are designed for individuals and pairs respectively. In the inference of SP for individuals, a weighted matrix factorization is utilized to fill the missing ratings. SP between an individual and a room is replaced by the satisfaction between an individual and the category. In the inference of SP for pairs, the community is detected and the satisfaction is inferred based on pairs with known ratings in the same community.

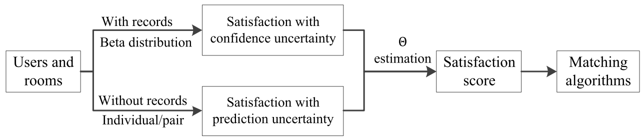


FIGURE 2. The framework of the proposed scheme.

Afterwards, $[\theta_1, \theta_2]$ is learned by considering the influences of confidence and prediction uncertainty. At last, the matching problem is proved to be NP-Hard. With all the elements calculated in G, two greedy matching algorithms are proposed to achieve optimal objects.

IV. SATISFACTION INFERENCE WITH UNCERTAINTY

For SC, the history records are regarded as direct evidence for inference. The pairs have an exponentially growing quantity and more sparse records than an individual. Thus, two different algorithms are designed for individuals and pairs in SP inference.

A. SATISFACTION WITH CONFIDENCE UNCERTAINTY

Given the check-in history records, the satisfaction between users and rooms is calculated with confidence uncertainty. Bayesian inference is a statistical inference approach to calculate probability based on evidence or observations. The normalized variance of the Beta distribution is utilized to calculate SC as follows [20].

$$\begin{cases} P_u = \dfrac{12\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \\ P_h = \dfrac{\alpha}{\alpha+\beta}(1-P_u) \\ P_d = \dfrac{\beta}{\alpha+\beta}(1-P_u) \end{cases} \tag{4}$$

	i_1	i_2	i_3	i_4
u_1	5(2)	3(1)	?	1(2)
u_2	4(3)	?	?	1(1)
u_3	1(1)	1(1)	?	5(2)
u_4	1(2)	?	?	4(4)
u_5	?	1(2)	5(2)	4(2)

(a)

	i_1	i_2	i_3	i_4
u_1	5(2)	3(1)	5.00(1)	1(2)
u_2	4(3)	1.96(1)	4.69(1)	1(1)
u_3	1(1)	1(1)	2.45(1)	5(2)
u_4	1(2)	0.46(1)	2.00(1)	4(4)
u_5	3.40(1)	1(2)	5(2)	4(2)

(b)

	c_1	c_2
i_1	1	0
i_2	1	0
i_3	0	1
i_4	0	1

(c)

	c_1	c_2
u_1	[0.31 0.48 0.20]	[0.21 0.48 0.31]
u_2	[0.41 0.38 0.21]	[0.20 0.60 0.20]
u_3	[0.14 0.45 0.41]	[0.31 0.48 0.21]
u_4	[0.14 0.32 0.54]	[0.50 0.3 0.20]
u_5	[0.21 0.48 0.31]	[0.63 0.24 0.13]

(d)

	i_1	i_2	i_3	i_4
u_1	[0.41 0.45 0.14]	[0.11 0.67 0.22]	..	..
u_2	[0.54 0.32 0.14]	[0.22 0.57 0.21]	..	..
u_3	[0.11 0.67 0.22]	[0.11 0.67 0.22]	..	..
u_4	[0.14 0.45 0.41]	[0.06 0.81 0.13]	..	..
u_5	[0.08 0.72 0.20]	[0.14 0.45 0.41]	..	..

(e)

FIGURE 3. Example of the individual-room satisfaction inference. (a) Individual-room rating matrix. (b) Filled missing data with weighted MF. (c) Room-category matrix. (d) Individual-category satisfaction matrix. (e) Individual-room satisfaction matrix.

where α and β represent the evidence of happy and unhappy nights staying in the room, respectively. To start, each user in the network has a prior observation, $\alpha = \beta = 1$. With the increase of observations, there will be less uncertainty for the inference. In this case, 3-stars is the threshold for distinguishing happiness and unhappiness for 0-5 star rating system. The nights are added to α or β. For example, if the rating is greater than 3-stars, the nights spent are added to α.

Fig. 3 gives an example of the satisfaction inference between individuals and rooms. As shown in Fig. 3(a), u_1 stayed in i_1 for 2 nights and highly recommended this room with 5 stars. Thus, $IRS[1][1]$ is calculated using Eq.4 with the confidence uncertainty. $\alpha = 2 + 1 = 3$, $\beta = 1$, $P_u = 12*3*1/(3+1)^2/(3+1+1) = 0.45$, $P_h = 3/(3+1)*(1-0.45) = 0.41$, $P_d = 1/(3+1)*(1-0.45) = 0.14$. As a result, $IRS[1][1]$=[0.41 0.45 0.14] as shown in Fig. 3(e).

B. INDIVIDUAL SATISFACTION WITH PREDICTION UNCERTAINTY

Since users have similar ratings to the rooms of a same category, a user's *SP* for a room is replaced by their satisfaction with the category. Individuals' check-in records and rooms in certain categories are evidence. We first divide rooms into several categories using information entropy. Afterwards, the missing ratings are inferred using weighted matrix factorization algorithm. Trust level models the prediction biases between individual-room satisfaction and individual-category satisfaction. *SP* is calculated based on ratings and the trust level for each category.

1) CATEGORY DIVISION

The relevant and valuable attributes of rooms are selected to constitute the categories. We utilize an attribute extraction algorithm in [21] to obtain top informative attributes. Shannon entropy [22] is utilized to quantify the expected informative value in attribute.

$$E(a_i) = -\sum_{j=1}^{N_{a_i}} p(x_j) log_2 p(x_j), \quad i \in [1, N_a] \tag{5}$$

where $E(a_i)$ denotes the entropy of the attribute a_i, and p denotes a_i's probability mass function. $\{x_1, \ldots, x_{a_i}\}$ is the value set for attribute a_i. N_a denotes the number of attributes. N_{a_i} denotes the number of values in a_i.

2) WEIGHTED MATRIX FACTORIZATION ALGORITHM

To fill the missing data in the *IRR* matrix for an individual, the weighted Matrix Factorization algorithm is utilized to infer the ratings. The existing cells are regarded as a training set. A user's rating of a room is represented as a couple of latent factors $\{V_1, V_2\}$. The rating matrix can be approximated as $R = \langle V_1, V_2 \rangle$. Considering the influence of the length of the nights stayed, L, the optimization problem will be adjusted as

$$\min_{U,V} ||(R - \langle V_1, V_2 \rangle) \odot L||_F^2 + \lambda_{v_1}||V_1||_F^2 + \lambda_{v_2}||V_2||_F^2 \tag{6}$$

where the number of nights stayed is regarded as a weight, $||\odot||_F^2$ is a Frobenius norm, and $\odot$ is a Hadamard product for element to element multiplication. To solve this problem, a batch gradient descent (BGD) is used to learn the latent factors V_1 and V_2 [7]. The number of nights spent in the room is are set as 1 for the absent records.

3) TRUST LEVEL CALCULATION

Considering the bias in Matrix Factorization, the individual's trust level for a category is also modeled as $\{belief, trustuncertainty, disbelief\}$, represented as $\{T_b, T_u, T_d\}$. Similar to {*happiness, uncertainty, unhappiness*}, the trust level is calculated considering the bias. The trust level measures the relationship between *IRS* and *ICS*. We present the algorithm for the trust level based on a Leave-one-out cross validation (LOOCV) [23]. For a category with n check-in records for an individual, n experiments are performed. For each experiment, one record is selected as the test data and the remaining ratings are utilized for training. If the experiment results and the ground truth value indicate the same preference (happy or unhappy), this record is regarded as belief evidence and L is added to the belief value α_t. Otherwise, L is added to the disbelief value β_t. The trust level is calculated in Eq.4 with $\{T_b, T_u, T_d\}$.

4) *SP* CALCULATION

For $ICS[i][j]$, the nights stayed L_{ik} with $IRR[i][k] > 3$ for each k is added to the happy evidence α', where the k_{th} room belongs to the j_{th} category. The summation of the nights stayed with low ratings is regarded as β' in the category. Based on Eq.4, $ICS[i][j]$ is calculated and is represented as $\{P'_h, P'_u, P'_d\}$. With an individual's trust level for a category, *SP* is calculated using Eq.5. P'_h and P'_d in *ICS* are converted to P_h and P_d in *IRS* with a proportion of T_b. T_d and T_u become two parts of P_u in *IRS*.

$$\begin{cases} P_h = T_b * P'_h \\ P_d = T_b * P'_d \\ P_u = T_b * P'_u + T_d + T_u \end{cases} \tag{7}$$

Fig. 3 gives an example of the *SP*'s inference for individuals and rooms. As shown in Fig. 3(a), u_5 has never stayed in i_1 so its ratings is unknown. The missing data is predicted using the weighted MF, as shown in Fig. 3(b), and the room category is divided as shown in Fig. 3(c). For $ICS[5][1]$, $\alpha = 1 + 1 = 2$, $\beta = 1 + 2 = 3$, $P'_u = 12 * 2 * 3/(2 + 3)^2/(2 + 3 + 1) = 0.48$, $P'_h = 2/(2 + 3) * (1 - 0.48) = 0.21$, and $P'_d = 3/(2 + 3) * (1 - 0.48) = 0.31$. Considering the trust level [0.41 0.45 0.14] between the individual and category, $IRS[5][1]$ is adjusted based on $ICS[5][1]$, and it equals [0.08 0.72 0.20], as shown in Fig. 3(e).

The process of *SC* and *SP* calculation between individuals and rooms is given as shown in Algorithm 1. As shown in the example, we can see that u_3, u_4, and u_5 have similar preference based on the existing ratings in Fig. 3(a). They create similar satisfaction trends with different uncertainty values, as shown in Fig. 3(e). u_4 stays longer in i_1 than u_3 does. As a result, u_4 shows less confidence uncertainty than u_3 in the inference.

Algorithm 1 Calculate Individual-Room Satisfaction

Input: *IRR* Matrix with missing data, Category matrix *C*
Output: *IRS* matrix

for Each existing cell in *IRR*[i][j] **do**
  if Rating R_{ij} is more than 3 stars **then**
    The nights stayed L_{ij} is added to α
  else
    L_{ij} is added to β
  end if
  Calculate $IRS[i][j]$ with confidence uncertainty
end for
Divide the rooms into several categories
Fill the missing data in *IRR* using weighted MF
Calculate each cell in *ICS* matrix
for Each missing cell in *IRS*[i][j] **do**
  Find j_{th} room's category k
  Assign $IRS[i][j] = ICS[i][k]$
end for
return *IRS* matrix

C. PAIR SATISFACTION WITH PREDICTION UNCERTAINTY

With n individuals in a user set, there will be a matrix of $n(n - 1)/2$ pairs. Due to the exponential growth of the matrix size, the pair-room rating records are much sparser than the individuals'. With the increasing number of pairs, there is a heavy overhead (especially in LOOCV) if utilizing the *SP* inference algorithm for individuals. With the existing check-in records, pairs' *SC*s are inferred in advance.

Algorithm 2 Calculate Pair-Room Satisfaction

Input: *IRR* Matrix, *PRR* Matrix
Output: *PRS* matrix

Calculate *SC* using pre-known ratings in *PRR* and fill in *PRS*
Detect the user community based on ratings in *IRR*
for Each missing cell in $PRS[i, j][k]$ **do**
  if $\{i, j\}$ belongs to the same community c_1 **then**
    Search for pairs $\{m, n\}$ with $\{i, j, m, n\} \in c_1$ and $PRS[m, n][k]$ is known
    Calculate the trust level between $\{i, j\}$ and $\{m, n\}$
    Synthesize satisfaction with pre-known pairs' satisfaction and trust levels
  else
    Search for pairs $\{m, n\}$ with $\{i, m\} \in c_1$, $\{j, n\} \in c_2$ and $PRS[m, n][k]$ is known
    Calculate the trust level
    Synthesize prediction satisfaction
  end if
end for
return *PRS* matrix

The *SP* between a pair and a room is calculated using a pre-known pair's *SC* and the trust levels between the pairs as show in Algorithm 2.

1) COMMUNITY DETECTION

Unlike the *SP* inference algorithm for individuals, a community-based inference algorithm is proposed for the *SP* inference algorithm for pairs. According to the *IRR* matrix with filled data, the community is detected based on rating similarities. A simple and fast algorithm for the K-means clustering algorithm [24] is utilized to cluster users into k communities.

Assuming that $PIR[i, j][k]$ is predicted with all the pre-known $PRS[m, n][k]$, $\{m, n\}$ are limited to the pairs that have the same community information as $\{i,j\}$. If i and j belong to the same community c_1, the pairs $\{m, n\}$ that all belong to c_1 are selected. If i and k belong to two different communities c_1 and c_2, the pairs $\{m, n\}$ in which m and n belong to c_1 and c_2 respectively are selected for prediction.

2) TRUST LEVEL CALCULATION

For pairs $\{i, j\}$ and $\{m, n\}$, the rating performance of four combinations $\{i, m\}$, $\{i, n\}$, $\{j, m\}$, and $\{j, n\}$ are utilized to evaluate the pairs' trust levels. Based on the two users' ratings in *IRR*, the same attitudes (both happy or both unhappy) are regarded as a belief observation. Otherwise, it is a disbelief observation. The counterpart observations in the four combinations are added up as belief evidence α_t and disbelief evidence β_t. If i=m, the pair's trust level equals the trust level between j and n.

3) *SP* CALCULATION

Based on the detected communities i and j, n pairs with a pre-known satisfaction with room k are selected to infer

$PRR[i,j][k]$. Let $\{P'_h, P'_u, P'_d\}$ represent confidence satisfaction $PRS[m,n][k]$. The prediction satisfaction is calculated using the pre-known pairs' satisfactions and trust levels with uncertainties as shown below. To synthesize the prediction from different pairs to one particular pair, a simple method is used to calculate the average.

$$\begin{cases} P_h = \frac{1}{n}\sum_{i=1}^{n} T_b^i * P_h'^i \\ P_d = \frac{1}{n}\sum_{i=1}^{n} T_b^i * P_d'^i \\ P_u = \frac{1}{n}\sum_{i=1}^{n} \{T_b^i * P_u'^i + T_d^i + T_u^i\} \end{cases} \tag{8}$$

For example, according to the ratings in Fig. 3(b), there are 4 belief observations between u_1 and u_2. Thus, $\alpha = 5$ and $\beta = 1$. The trust level with the uncertainty equals $\{0.63, 0.24, 0.13\}$. The trust level between $\{u_1, u_3\}$ and $\{u_2, u_4\}$ equals $\{0.52, 0.15, 0.33\}$. For room k, if only $\{u_1, u_3\}$ is selected with $PRS[u_1, u_3][k]$=$\{0.5, 0.3, 0.2\}$, $PRS[u_2, u_4][k] = \{0.26, 0.575, 0.165\}$.

V. BIPARTITE MATCHING

A. TWO GREEDY ALGORITHMS FOR MATCHING

The matching problem between users and rooms is NP-Hard based on a reduction from the NP-hard Maximum Weight Independent Set (MWIS) problem [18], [25]. Fig. 4 illustrates an example of the reduction from MWIS to the matching problem. The blue node with grid represents a pair of users, while the yellow node represents an individual user. The numbers in the brackets represent the weights considering the utility and satisfaction. As shown in Fig. 4(b), the nodes in a circle indicate a pair of conflicted users, who can not be assigned to the same room. u_1 and $\{u_1, u_2\}$ cannot be matched to i_1 at the same time. Based on the hardness of the approximation result for MWIS [26], the matching problem is not approximated with $n^{1-\varepsilon}$ for any $\varepsilon > 0$.

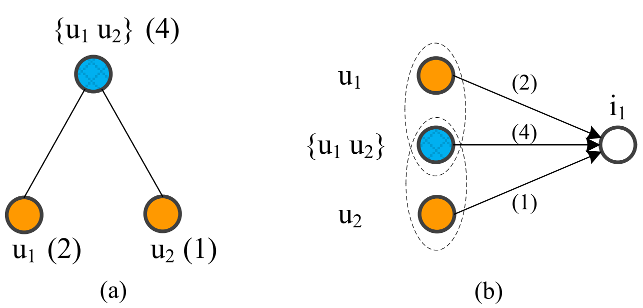


FIGURE 4. Example of reduction from MWIS to matching problem. (a) A MWIS instance. (b) A matching instance.

Considering the room assignment rules in the matching process, we present two greedy algorithms. The basic idea is described as follows. The maximum matching is a matching with the largest possible U_{all}, no matter whether each user is assigned to a room. The principle of maximal matching is to pursue the largest possible U_{all} while ensuring that each user has a room.

1) ALGORITHM FOR MAXIMUM MATCHING

We present a greedy algorithm for maximum matching as shown in Algorithm 3. The weights are the products of a user's utility and an edge's score. A hyper-edge represents a matching from a pair to a room. After sorting the edges, the edge with the largest weight is added to construct subgragh H. We delete the conflict edges in graph G until the graph is empty. For unmatched users, one user is added to a double rooms in each round by the sorted increased rating to U_{all}. If the increase is negative, the unmatched users are abandoned.

Algorithm 3 A Greedy Algorithm for Maximum Matching

Input: Graph G, G=$(U \cup R, E, Score, Q)$

Output: Subgraph H, U_{all}

```
1:  Calculate the weights for each edge e, U_all = 0
2:  Sort all edges from largest to smallest based on weights
3:  for Egdes in G do
4:      Add the edge with largest weight to subgragh H
5:      if the e is hyper-edge, denoted as e({i, j}, k) then
6:          Delete all the edges with vertex i and j in G
7:          Add two times the weight to U_all
8:      else
9:          the e is denoted as e(i, k)
10:         Delete all the edges with vertex i in G
11:         Add the weight to U_all
12:     end if
13: end for
14: Add the unmatched users to vehicles with only one individual
15: Return Subgraph H and U_all
```

This average-based algorithm is highly suitable for situations in which the quantity of rooms are no less than the quantity of users. The algorithm shows high scalability in practice.

2) ALGORITHM FOR MAXIMAL MATCHING

As shown in Algorithm 4, a maximal matching is a matching which no more edges can be added. It is a local maximum. n, m_1, and m_2 represent the quantity of users, a single room, and a double room, respectively. To ensure the principle, there are at least $(n - m_1)/2$ double rooms which are required to accommodate each pair of users. The different value refers to the gap between the pair's maximal weight in all rooms and the sum of two individual weights. For example, the pair's maximal weight is the largest weight of $e(\{i,j\},k), \forall k, 1 \le k \le m_1 + m_2$. Two individuals's maximal weight is the largest weight of $e(i,k)$ and $e(j,k), \forall k, 1 \le k \le m_1 + m_2$. We first match $(n - m_1)/2$ pairs to rooms with the sorted difference value from the smallest to the largest. If the difference value is negative, it means the pair of users gets along well, which and has a positive impact on the accommodation. After each round, the difference and sorted list is updated. Algorithm 3 (Lines 2-13) is utilized for the remaining rooms.

If the quantity of users is no greater than that of the rooms, the result of the maximum matching is the same as that of the maximal matching. The algorithm takes individual matching into consideration first and pursues the optimal result for the rest users.

Algorithm 4 A Greedy Algorithm for Maximal Matching

Input: Graph G, $G=(U \cup R, E, Score, Q)$
Output: Subgraph H, U_{all}

1: Calculate the weights for each edge e, $U_{all} = 0$
2: $n = |U|$; $m_1 = \sum_{i \in I} 1, \forall Q(i) = 1$; $m_2 = |I| - m_1$
3: $m_{min} = (n - m_1)/2$
4: **for** $m_{min} \geq 1$ **do**
5: Calculate the differences between hyper-edges and edges
6: Sort the hyper-edges from smallest to largest
7: Add the top hyper-edge $e(\{i, j\}, k)$ to subgragh H
8: Delete all the edges with vertex i and j in G
9: Add two times the weight to U_{all}, $m_{min} = m_{min} - 1$
10: **end for**
11: Continue using **Algorithm 3**
12: Return Subgraph H and U_{all}

B. ESTIMATION OF PARAMETER θ

In this subsection, we present an algorithm to estimate θ, which models the different impacts of confidence and prediction uncertainty. To evaluate the performance of the $\{\theta_1, \theta_2\}$ setting, we utilize a popular and easy-to-use method to minimize the error with Gradient Descent. The graph G is regarded as the input and U_{all} is the output. In the training data, the scores for the edges with the ground truth are set as 1 with an expected output of U^*_{all}. The distance between U_{all} and U^*_{all} is denoted as

$$d(U_{all}, U^*_{all}) = \frac{1}{2}(U^*_{all} - U_{all})^2 \tag{9}$$

With a set of N samples, the loss function is defined as

$$loss(\theta) = \sum_{i=1}^{N} d(U_{all}, U^*_{all}) \tag{10}$$

The loss function is utilized to evaluate the quality of θ, denoted as

$$\widehat{\theta} = \arg\min_{\theta} loss(\theta) \tag{11}$$

As an iteration method, Gradient Descent sets a random value for θ and is adjusted by iterations to decrease the value of the loss function. When the value of the loss function is minimized, θ performs best in the training data. As a result, this set of θ is selected in the test data.

VI. EXPERIMENT EVALUATIONS

We conduct extensive experiments based on two real-world datasets, and then we present a comprehensive evaluation of our methods.

A. METHODOLOGY

We test our methods on two real-world datasets, Airbnb [27] and Happymovie [28] (an extension of the Movielens dataset [29]). Airbnb is a worldwide platform for hosts and users sharing a house space. We select records in New York City, which has 40,227 listings that contain rich room attributes. Movielens has 1,000,209 ratings (1-5 stars), 3,900 movies, and 6,040 users, and the Happymovie dataset selects 58 users, 50 movies, and some additional group recommendation information.

In the following, we give the evaluation and experiment results. First, we experiment with the real-world datasets in terms of satisfaction inference, and we show the effectiveness of the proposed algorithm. Before performing the matching algorithm on the datasets, we need to generate the evaluation model. In the satisfaction inference evaluation, the confidence uncertainty and prediction uncertainty for individuals and pairs are evaluated separately. The performance of the matching algorithms are given afterwards.

B. EVALUATION OF SATISFACTION INFERENCE

In the Happymovie dataset, movies are divided to 18 categories based on themes. The rooms in the Airbnb dataset can be divided into several kinds of categories based on different attributes. As shown in Table 2, the information entropy of attributes is calculated. The relevant and valuable attributes with the top entropy of rooms are selected to constitute the categories.

TABLE 2. Information entropy of attributes in Airbnb database.

Attribute	Entropy	Attribute	Entropy
Room_type	1.18	Cancellation_policy	1.54
Price	2.41	Neighbourhood_group	1.48
Minimum_nights	1.89	Review_rate	2.20

Due to privacy issues, the check-in records of pairs are hard to acquire from the Airbnb datasets. Happymovie dataset has both pair and group information. As a result, we focus on using the Happymovie dataset in our evaluation. We run all programs in Windows 10 on a computer with an Intel(R) Core(TM) i7 CPU and 6 GB RAM.

In the following, we conduct experiments in satisfaction inference with the Happymovie dataset, and we show the effectiveness of the proposed algorithm. In the satisfaction inference evaluation, confidence uncertainty and prediction uncertainty for individuals and pairs are evaluated separately. The performance of the matching algorithm is given afterwards.

1) *SC* EVALUATION

For satisfaction with confidence uncertainty, Fig. 5 illustrates the probability distribution of happiness, uncertainty, and unhappiness. With more nights spent in the rooms, the uncertainty will decrease.

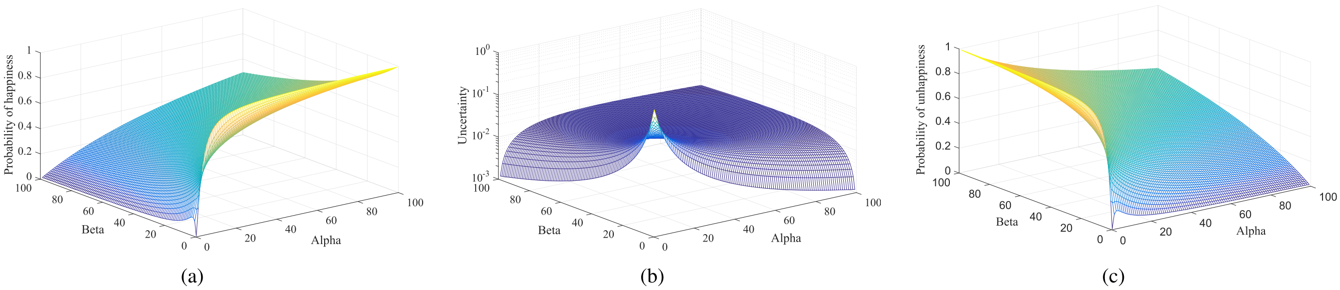


FIGURE 5. Evaluation for relationship between confidence uncertainty and α, β. (a) Probability of happiness. (b) Probability of uncertainty. (c) Probability of unhappiness.

2) INDIVIDUAL'S *SP* EVALUATION

We use the RMSE metric to evaluate the performance of a weighted MF. RMSE is defined as

$$RMSE = \sqrt{\sum_{R_{i,j} \in D^{test}} \frac{(R_{i,j} - \widetilde{R}_{i,j})^2}{|D^{test}|}} \tag{12}$$

where $R_{i,j}$ is the predicted value using the weighted MF and D^{test} is the test data. In the algorithm, λ is a constant that determines the rate at which we approach the minimum. μ is used to control the magnitudes of the user-feature and item-feature vectors.

First, we experiment to select the proper parameters for improving the inference performance. As shown in Fig. 6(a), we divide the test data and training data with different percentages. In practice, we set λ as 0.0002 and μ as 0.02 with 500 steps. The RMSE shows a positive correlation with the test data percentages in all datasets.

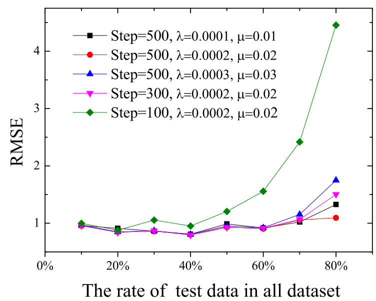


FIGURE 6. Comparison of RMSE metric in different test data rate.

As presented in [6], neural network models are utilized in Airbnb website. We use the model proposed in [30] as baseline. As shown in Fig. 7, our work outperforms the baseline in prediction.

Fig. 8 shows the performance of the prediction uncertainty in accuracy and uncertainty. In [8, Scheme 1], the error caused by MF is not considered. Our work shows more improvement than Scheme 1. With the growing percentage of test data, the uncertainty is increased.

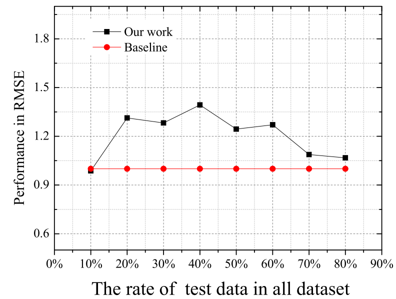


FIGURE 7. Comparison of RMSE metric in different algorithms.

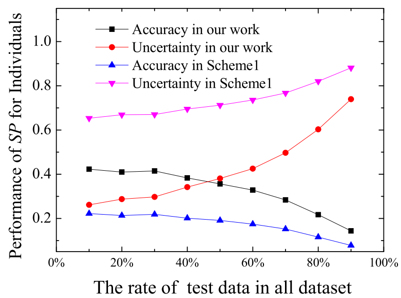


FIGURE 8. Performance of the individual prediction uncertainty algorithm.

3) PAIR'S *SP* EVALUATION

In the dataset, the users are divided into several groups and rate the movies. We generates the ground truth of the pair-room-ratings based on rankings and the rating distance between two individuals in a pair. When two individuals have similar rating preference, an extra points are added to the average individual rating as pair-room-rating.

Fig. 9 shows the relationship between the trust level and the community quantity. The users inside the community show both the highest and lowest beliefs when k=8 in the dataset.

Fig. 10 shows the performance of the salification inference with prediction uncertainties in accuracy and uncertainty. With the increased density of the records, the accuracy grows and the uncertainty is reduced.

C. EVALUATION OF MATCHING ALGORITHMS

In the evaluation of the matching algorithms, the product value of the scores between users and rooms is generated

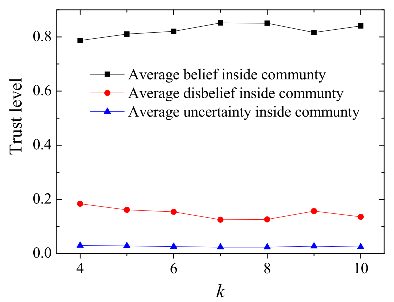


FIGURE 9. Relationship between trust level and community number.

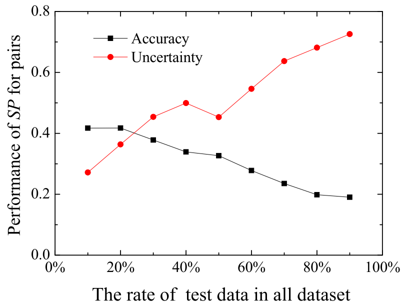


FIGURE 10. Performance of the individual prediction uncertainty algorithm.

randomly using a uniform distribution. Regardless of user and room conflicts, we sum all highest values of each user as the upper bound. The optimal result is acquired using matching algorithms. The ratio between the optimal result and the upper bound is regarded as the measure of evaluating the matching algorithm. We select different rates of the room quantity and the user quantity. As shown in Fig. 11, the measure ratio reaches the lowest point when the maximum amount of users the rooms can accommodate equals the quantity of users. If more users or rooms remain after matching, the matching results approach the upper bound.

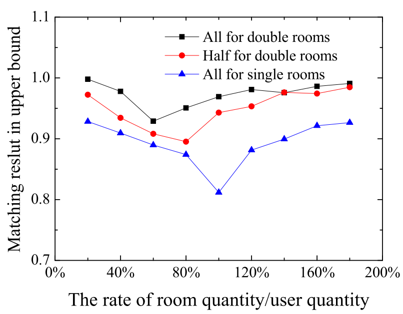


FIGURE 11. The influence of the room/user rate.

As shown in Fig. 12, the quantity of users and rooms in case 1 are five times of that in case 2. With the increase of user and room quantities, the optimal results of the matching algorithms are more likely to approach the upper bound. If the number of users is more than that of the rooms, the maximal matching is more likely to perform better. Otherwise, if the number of users is limited, the maximal matching is more likely to perform better. This average-based algorithm is highly suitable for situations in which the quantity of rooms are no less than the quantity of users.

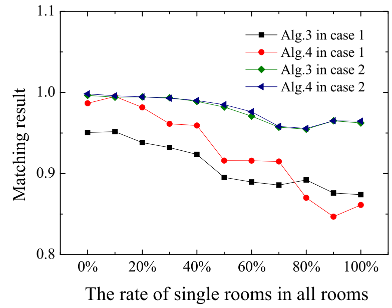


FIGURE 12. Performance between two algorithms.

VII. DISCUSSION

In the proposed model, a matching problem under uncertainty is solved based on inferred satisfaction. The proposed method takes various factors into consideration as follows.

- Confidence uncertainty and prediction uncertainty. Two uncertainties are caused by data confidence and prediction error, respectively. The weights of these two kinds of uncertainty are set using a learning algorithm.
- Individual and pair. With different data sparse and data scales, two algorithms are designed for two different cases. In the base of individuals' rating inference, the pairs' satisfaction is inferred.
- Integrated matching model. The proposed model integrates all the metrics above with improved accuracy and efficiency.

In pair satisfaction inference, we utilise a rating preference-based algorithm to detect communities with the traditional clustering algorithm k-means. With the development of clustering and community detection technologies, the community detection will be more realistic and contributes to the accuracy of the pair satisfaction inference.

With the inferred satisfaction, the problem can eventually be reduced to a bipartite matching problem that has been proved to be NP-Hard.

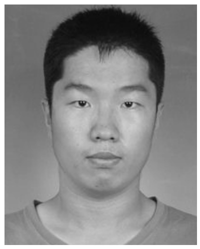
VIII. CONCLUSION AND FUTURE WORK

In this paper, user salification with rooms is calculated using the influence of confidence and prediction uncertainties. An individual's satisfaction with the prediction uncertainties is modeled using a weighted matrix factorization-based algorithm. A pair's satisfaction with the prediction uncertainties is modeled based on pairs' similarity in a community. The utility and uncertainty objectives are combined in a single objective and are optimized using a matching algorithm. Finally, extensive experiments using real-world data show the effectiveness and accuracy of the proposed method.

With the development of clustering and community detection technologies, community detection will be more realistic and will contribute more to the accuracy of pair satisfaction inference. Taking advantage of social information, we can delete the hyperedges of the pairs that are not suitable to occupy the same room. The decreased quantity of hyperedges contributes to the improvement of efficiency and the reduction of complexity. In real life, users with a higher social influence will cause more reputation influence to the platform. If the utility of users is considered, the model will be more realistic.

REFERENCES

[1] Z. Fang, L. Huang, and A. Wierman, "Prices and subsidies in the sharing economy," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 53–62.
[2] X. Ma, J. T. Hancock, K. L. Mingjie, and M. Naaman, "Self-disclosure and perceived trustworthiness of airbnb host profiles," in *Proc. CSCW*, 2017, pp. 2397–2409.
[3] W. Chang and J. Wu, "Social VoD: A social feature-based P2P system," in *Proc. 44th Int. Conf. Parallel Process. (ICPP)*, Sep. 2015, pp. 570–579.
[4] T. Ikkala and A. Lampinen, "Monetizing network hospitality: Hospitality and sociability in the context of airbnb," in *Proc. 18th ACM Conf. Comput. Supported Cooper. Work Social Comput.*, 2015, pp. 1033–1044.
[5] E. Ert, A. Fleischer, and N. Magen, "Trust and reputation in the sharing economy: The role of personal photos in Airbnb," *Tourism Manage.*, vol. 55, pp. 62–73, Aug. 2016.
[6] M. Grbovic, "Search ranking and personalization at Airbnb," in *Proc. 11th ACM Conf.*, 2017, pp. 339–340.
[7] X. Wang, W. Pan, and C. Xu, "HGMF: Hierarchical group matrix factorization for collaborative recommendation," in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage.*, 2014, pp. 769–778.
[8] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.
[9] H. Zhu, H. Xiong, Y. Ge, and E. Chen, "Mobile APP recommendations with security and privacy awareness," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 951–960.
[10] H. Li, Y. Ge, R. Hong, and H. Zhu, "Point-of-interest recommendations: Learning potential check-ins from friends," in *Proc. KDD*, 2016, pp. 975–984.
[11] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*. New York, NY, USA: Springer, 2011.
[12] Y. Zhang, W. Zhang, X. Lin, M. A. Cheema, and C. Zhang, "Matching dominance: Capture the semantics of dominance for multi-dimensional uncertain objects," in *Proc. 26th Int. Conf. Sci. Statist. Database Manage.*, 2014, p. 18.
[13] C. Long, R. C.-W. Wong, P. S. Yu, and M. Jiang, "On optimal worst-case matching," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2013, pp. 845–856.
[14] I. Molloy, L. Dickens, C. Morisset, P.-C. Cheng, J. Lobo, and A. Russo, "Risk-based security decisions under uncertainty," in *Proc. 2nd ACM Conf. Data Appl. Security Privacy*, 2012, pp. 157–168.
[15] N. Osman, C. Sierra, F. Mcneill, J. Pane, and J. Debenham, "Trust and matching algorithms for selecting suitable agents," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 1, p. 16, 2013.
[16] W. Zhang, J. Wang, B. Chen, and X. Zhao, "To personalize or not: A risk management perspective," in *Proc. 7th ACM Conf. Rec. Syst.*, 2013, pp. 229–236.
[17] H. Zhao, Q. Liu, G. Wang, Y. Ge, and E. Chen, "Portfolio selections in P2P lending: A multi-objective perspective," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 2075–2084.
[18] C. Chen, S. Chester, V. Srinivasan, K. Wu, and A. Thomo, "Group-aware weighted bipartite B-matching," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 459–468.
[19] Y. Sasaki *et al.*, "The truth of the F-measure," *Teach Tutor mater*, vol. 1, no. 5, pp. 1–5, 2007.
[20] F. Li and J. Wu, "Uncertainty modeling and reduction in manets," *IEEE Trans. Mobile Comput.*, vol. 9, no. 7, pp. 1035–1048, Jul. 2010.
[21] J. Wu and Y. Wang, "Social feature-based multi-path routing in delay tolerant networks," in *Proc. IEEE Infocom*, Mar. 2012, pp. 1368–1376.
[22] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 5, no. 1, pp. 3–55, 2001.
[23] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. Int. Joint Conf. AI*, Aug. 1995, vol. 14. no. 2, pp. 1137–1145.
[24] A. Vattani, "K-means requires exponentially many iterations even in the plane," *Discrete Comput. Geometry*, vol. 45, no. 4, pp. 596–616, 2011.
[25] A. W. Kolen, J. K. Lenstra, C. H. Papadimitriou, and F. C. Spieksma, "Interval scheduling: A survey," *Naval Res. Logistics*, vol. 54, no. 5, pp. 530–543, 2007.
[26] J. Håstad, "Clique is hard to approximate withinn1- ε," *Acta Mathematica*, vol. 182, no. 1, pp. 105–142, 1999.
[27] Accessed: Feb. 11, 2018. [Online]. Available: http://insideairbnb.com/
[28] Accessed: Feb. 11, 2018. [Online]. Available: http://gaia.fdi.ucm.es/research/happymovie/download/
[29] Accessed: Feb. 11, 2018. [Online]. Available: https://grouplens.org/datasets/movielens/
[30] J. C. Martineau, D. Cheng, and T. Finin, "TISA: Topic independence scoring algorithm," in *Proc. Int. Conf. Mach. Learn. Data Mining Pattern Recognit.*, 2013, pp. 555–570.



LONGHUA GUO received the B.S. degree in electronic information engineering from Tianjin University, Tianjin, China, in 2013, and is currently pursuing the Ph.D. degree with Shanghai Jiao Tong University, Shanghai, China. He participated in many national projects, such as National Natural Science Foundation of China and National 973 Planning of the Ministry of Science and Technology, China. His research interests include sensor network security, and social network analysis. He was a Visiting Scholar with Temple University from 2016 to 2017.

JIANHUA LI received the B.S., M.S., and Ph.D. degrees from Shanghai Jiao Tong University, in 1986, 1991, and 1998, respectively. He is currently a Professor, a Ph.D. Supervisor, and the Dean of the School of Cyber Security, Shanghai Jiao Tong University, Shanghai, China. He is also the Director Expert of Information Security Committee of National High Technology Research and Development Program of China (863 Program). He was the Leader of over 30 state/province projects of China, and authored or co-authored over 200 papers. He published six books and has about 20 patents. He made three standards and has five software copyrights. His research interests include information security, signal process, and computer network communication. He is the member of the Committee of Information Security Area of the State Tenth Five-Year Plan of China. He is also a Committee Expert of China State Secrecy Bureau, Shanghai Secrecy Bureau, and Information Technique Standardization Committee of Shanghai, China. He received the First Prize of National Technology Progress Award of Shanghai in 2003 and 2004, two First Prizes of National Technology Progress Awards of Shanghai in 2004, and the Second Prize of National Technology Progress Award of China in 2005.

JIE WU (F'09) was the Program Director with the National Science Foundation and was a Distinguished Professor with Florida Atlantic University. He is currently the Chair and a Laura H. Carnell Professor with the Department of Computer and Information Sciences, Temple University. He is also an Intellectual Ventures endowed Visiting Chair Professor with the National Laboratory for Information Science and Technology, Tsinghua University. He regularly publishes in scholarly journals, conference proceedings, and books. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. He serves on several Editorial Boards, including the IEEE TRANSACTIONS ON SERVICE COMPUTING and the *Journal of Parallel and Distributed Computing*. He was a General Co-Chair/Chair for the IEEE MASS 2006, the IEEE IPDPS 2008, the IEEE ICDCS 2013, and the ACM MobiHoc 2014, and the Program Co-Chair for the IEEE INFOCOM 2011 and the CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visitor, an ACM Distinguished Speaker, and the Chair for the IEEE Technical Committee on Distributed Processing. He is a CCF Distinguished Speaker. He was a recipient of the 2011 China Computer Federation Overseas Outstanding Achievement Award.

WEI CHANG received the B.S. degree from the Beijing University of Posts and Telecommunications in 2009 and the Ph.D. degree in computer science from Temple University in 2016. He is currently an Assistant Professor with the Department of Computer Science, Saint Joseph's University, Philadelphia, PA, USA. His research interests include social information-assisted system design, and its related security and privacy issues. He serves as the Coordinator of Cybersecurity Program with the Computer Science Department, Saint Joseph's University.

JUN WU (M'11) received the Ph.D. degree in information and telecommunication studies from Waseda University, Japan, in 2011. He was a Post-Doctoral Researcher with the Research Institute for Secure Systems, National Institute of Advanced Industrial Science and Technology, Japan, from 2011 to 2012. He was a Researcher with the Global Information and Telecommunication Institute, Waseda University, from 2011 to 2013. He is currently an Associate Professor with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China. He is also the Vice Director with the National Engineering Laboratory for Information Content Analysis Technology, Shanghai Jiao Tong University. His research interests include the advanced computing, communications, and security techniques of software-defined networks, information-centric networks smart grids, and Internet of Things, where he has published over 90 refereed papers. He has been the TPC Member of over ten international conferences, including ICC, GLOBECOM, and WINCON. He is the Chair of the IEEE P21451-1-5 Standard Working Group. He has hosted and participated in a lot of research projects, including National Natural Science Foundation of China, National 863 Plan and 973 Plan of China, and Japan Society of the Promotion of Science Projects. He has been a Guest Editor of the IEEE SENSORS JOURNAL. He is an Associate Editor of the IEEE ACCESS.