2021 22nd IEEE International Conference on Mobile Data Management (MDM) | 978-1-6654-2845-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/MDM52706.2021.00028

2SRS: A Two-Sided Recommender System to Connect Local Businesses to Bus Passengers

Tahereh Arabghalizi
Department of Computer Science
University of Pittsburgh
e-mail: tahereh.arabghalizi@pitt.edu

Alexandros Labrinidis
Department of Computer Science
University of Pittsburgh
e-mail: labrinid@cs.pitt.edu

***Abstract*—Recommender systems are widely used to help customers find the most relevant and personalized products or services tailored to their preferences. However, traditional systems ignore the preferences of the other side of the market, e.g., "product suppliers" or "service providers", towards their customers. In this paper, we present *2SRS* a Two-Sided Recommender System that recommends coupons, supplied by local businesses, to passerby while considering the preferences of both sides towards each other. For example, some passerby may only be interested in coffee-shops whereas certain businesses may only be interested in sending coupons to new customers only. Our experimental results show that 2SRS delivers higher satisfaction when considering both sides of the market compared to the baseline methods.**

Keywords: Recommender Systems; Two-sided Markets; Mobile Applications

I. Introduction

Recommender systems have been extensively used in many real-world applications such as e-commerce and social media. With the rise of Smart City developments, urban recommendation solutions in the fields of energy, transportation, traffic, etc. have attracted much attention. In the *PittSmartLiving* project [1] we aim to build a marketplace around multimodal mobility, where businesses can offer time-sensitive incentives to nearby commuters utilizing transit information (e.g., the next bus is full, come in and enjoy $1 coffee). This has the potential to improve not only the overall ridership experience by balancing utilization across public transportation networks (e.g., shifting some of the demand away from the peak hours), but also to optimize customer flows in local businesses.

Problem Statement/Motivating Example: In this work we are addressing the problem of how to best recommend coupons (offered by local businesses located nearby) to bus passengers waiting for their bus to arrive. The coupons would be targeted for times when the next bus is expected to be full and encourage the (future) bus passengers to enjoy the recommended offer instead of trying to ride a full bus. We performed a survey in February 2017, using a diverse research registry of 3915 participants, out of which 891 responded within a 6-day period. One of the questions was: "Would monetary incentives (e.g., discount for coffee) help you decide in favor of waiting?". 47.8% of participants said YES to this question which shows the desire for a recommender system that meets this need. Such a recommender system is required to consider the preferences of both sides of the marketplace and recommend the most relevant coupons to the passengers whose attributes also satisfy the preferences of the local businesses.

Although our motivating application is focusing on people arriving at bus stops, our proposed solution is easily expandable to a much broader application space, which includes people walking around a city and receiving coupons on their mobile phone from nearby businesses. Our assumption (inspired by reality) is that the bus passengers arrive randomly at bus stops over time and the coupons associated with a bus stop also become randomly available over time. Therefore, our method offers the most relevant coupons, among the available ones at a bus stop, at the time when a bus passenger arrives, knowing the fact that their next bus is going to be full.

Related Work: Since the term *"Recommender System"* was first introduced in 1997 [2], many researchers have been proposing new approaches to improve the quality of the personalized recommendations created by content-based [3], collaborative filtering [4], hybrid methods [5], etc. Traditional $user \times item$ recommender systems provide items that satisfy only users' needs or interests and they are sufficient for many real-world applications. However, there exist other applications in which the user-centric approach is not enough and the preferences of all involved stakeholders need to be taken into account [6]. Reciprocal recommendation is a special case of the multi-sided recommender systems where the task is to match people to people. Reciprocal recommendation has some similarities to two-sided matching problems [7], however, in two-sided matching problems all matchings are exclusive and they are made at the same time. A reciprocal recommendation is successful if both parties accept it. Some examples of reciprocal recommendation systems include online dating, online recruiting, and two-sided sharing economy platforms such as Uber and AirBnB [8]–[10]. Another example of multi-sided recommender systems is used in two-sided marketplaces where suppliers of products or services are on one side and customers are on the other side of the market. The recommender system in such environment, is required to maximize the satisfaction of both stakeholders. However, most research in this area is concerned with enhancing provider fairness or diversity while preserving recommendation accuracy [11].

Contributions: In this paper, we propose a two-sided recommendation methodology that aims to fulfil the preferences

978-1-6654-2845-3/21/$31.00 ©2021 IEEE
DOI 10.1109/MDM52706.2021.00028

of both parties involved in the recommendation, e.g., bus passengers and local businesses. This system offers the top relevant items from suppliers to an incoming user whose attributes match the preferences of those suppliers. To the best of our knowledge, there is no recommendation solution for a two-sided marketplace that provides the relevance of recommendations to both users and suppliers. Unlike the reciprocal recommender systems, the success of a recommendation in our method depends only on the user's acceptance of an offer, although the recommendation should be also implicitly acceptable by the suppliers. We make the following contributions:

1) we propose 2SRS, a two-sided recommender system that takes the preferences of both sides of the market into consideration (Section II)
2) we present a model to simulate the arrival of the users and items (Section III).
3) we define two metrics to evaluate the satisfaction of both parties (Section IV).
4) we perform an extensive experimental evaluation and show that our proposed recommendation method surpasses the baselines (Section V, Section VI).

II. Proposed Recommendation Method

In this section, we describe 2SRS, a Two-Sided Recommender System that recommends items, provided by suppliers, to users. Our goal is to satisfy preferences of both parties towards each other such that a recommendation is acceptable to both sides of the marketplace. We consider the two-sided marketplace consisting of one set of Users U, one set of Items I and one set of Suppliers S. On one side of this marketplace, each user $u \in U$ has a set of self-descriptive attributes, A_u, and a set of preferred attributes towards the items, P_u. On the other side of the marketplace, each item $i \in I$ has a set of self-descriptive attributes, A_i, and the supplier $s_i \in S$ who provides this item has a set of preferred attributes towards the users, P_{s_i}. The goal of this approach is to recommend the most relevant items to each user whose attributes fit the preferred attributes of the suppliers of those items. In other words, user u and item i are considered a good match if and only if A_u satisfies P_{s_i} and A_i satisfies P_u. In order to determine a good match, we define *relevance(i,u)*, a function that computes the relevance between A_i and P_u and *relevance(u,i)*, a function that computes the relevance between A_u and P_{s_i}. These relevance scores are obtained using the cosine similarity of each two relevant feature vectors whose components correspond one by one (Eq. 1 and Eq. 2).

$$relevance(i,u) = \cos(A_i, P_u) \tag{1}$$

$$relevance(u,i) = \cos(A_u, P_{s_i}) \tag{2}$$

We define a two-sided relevance score which comprises the impact of both mentioned relevance scores and is calculated as the product of these two scores (Eq. 3). We use the operation of multiplication instead of a linear combination to remove the condition of one-sided preference. 2SRS computes the two-sided relevance score between each pair of user u and item i and then recommends the items that make the highest two-sided relevance scores with a user.

$$\begin{aligned} &two - sided\, relevance(u \leftrightarrow i) \\ &\quad = relevance(i,u) \times relevance(u,i) \end{aligned} \tag{3}$$

III. Evaluation Setup

We evaluated our proposed technique using a simulation model that is time-based and does not consider the location of the suppliers and the users. We assume both sides of the marketplace arrive and depart over time. We use two separate *Poisson Processes* with two different arrival rates to simulate the arrival of users and items. A Poisson Process generates events at random points of time at an average rate where no two events can occur at the same time [12]. We define λ_u as the user arrival rate and λ_i as the item arrival rate. If a random user arrives at time t, the amount of time until the next user arrival is computed using the following equation:

$$T = \frac{-\ln R}{\lambda} \tag{4}$$

where R is a random value between 0 and 1, λ is the user arrival rate and the next time is equal to $t+T$ [13]. The second Poisson process generates the arrival of items where a random item is populated by a supplier at time t' and the time until the next item becomes available is obtained by Eq. 4 where λ is the item arrival rate. According to this model, as soon as a user arrives (i.e., at time t), the recommender system identifies all items that are available at this time, considers them as the potential candidates for recommendation and offers the top candidates which have the highest two-sided relevance scores. In this model, the user accepts one of the recommended items at random which will be removed from the pool of the available items after it is accepted.

IV. Evaluation Metrics

As outlined earlier, our recommendation method aims to satisfy the preferences of both sides of the marketplace. Therefore, we define two metrics to measure the satisfaction of users and suppliers:

User Satisfaction: We define user satisfaction based on the mean of the ratings he/she would give to the recommended items. If a recommended item has been offered and rated by the user before, we use that rating to compute his/her satisfaction. Otherwise, we estimate the unknown rating using the mean of the ratings that similar users have given to that item. This is based on the fact that similar users have similar tastes and give similar ratings to the same items. In order to find the similar users, first we apply the K-Means algorithm to cluster the users based on their self-descriptive attributes (e.g., age group) and then specify the users who are in the same cluster as the similar users to a user. *User satisfaction* is a score between 0 and 5 that is obtained by Eq. 5 and 6:

$$User\ Satisfaction(u) = \sum_{i \in R_u} \frac{\phi(u,i)}{|R_u|} \tag{5}$$

TABLE I
BILATERAL FEATURES OF MOVIES AND USERS

	Movies' attributes/Users' preferred attributes	Users' attributes/Production companies' preferred attributes
Numerical	"year" => classic, 1970-2000, 2000s "duration" => short, medium, long "average vote" => low-rated, medium-rated, high-rated	"age" => 20-, 20-30, 30-40, 40-50, 50-60
Categorical	"genre", "language" (English, non-English)	"occupation", "watching status" (seen, unseen)

$$\phi(u,i) = \begin{cases} r(u,i), & \text{if } \ r(u,i) \neq None \\ \sum_{c \in C_u} \frac{r(c,i)}{|C_u|}, & \text{otherwise} \end{cases} \quad (6)$$

where $u \in U$ is a user, i is an item form the list R_u that is recommended to the user u, r(u,i) is the past rating that user u has given to item i, C_u are the similar users to user u who are in the same cluster and r(c,i) is the rating that user $c \in C_u$ has given to the item i. The total user satisfaction for each simulation is computed as the mean of the satisfactions of all users who received recommendations in that simulation.

Supplier Satisfaction: We define the satisfaction of a supplier based on how relevant the users are to the supplier's preferences. In particular, we can obtain the list of the users who are offered items from each supplier after each simulation. If the relevance score between a user and a supplier (Eq. 2) is greater than 0.8, we consider that user as a great match because his/her attributes satisfy the supplier's preferences the most and he/she gets a score of 5, if the relevance score is between 0.8 and 0.6, the user gets a score of 4 and so on. We can then compute the mean of the scores of the users who are offered items from a supplier and use it as the satisfaction score for that supplier. *Supplier satisfaction* is a score between 0 and 5 and computed using the following equations:

$$Supplier\ Satisfaction(s) = \sum_{u \in U_s} \frac{\theta(s,u)}{|U_s|} \quad (7)$$

$$\theta(s,u) = \begin{cases} 0, & \text{if } \ relevance(u,i) = 0 \\ 1, & \text{if } \ 0 < relevance(u,i) < 0.2 \\ 2, & \text{if } \ 0.2 <= relevance(u,i) < 0.4 \\ 3, & \text{if } \ 0.4 <= relevance(u,i) < 0.6 \\ 4, & \text{if } \ 0.6 <= relevance(u,i) < 0.8 \\ 5, & \text{if } \ relevance(u,i) >= 0.8 \end{cases} \quad (8)$$

where $u \in U_s$ ($U_s \subset U$) is a user from the list of users U_s who are recommended items from supplier s and relevance(u,i) is the relevance function that computes the relevance score between the user u and the supplier s who provides item i. The total supplier satisfaction of a simulation is calculated as the mean of the satisfactions of all suppliers involved in the simulation.

V. BASELINE METHODS

The recommendation baseline methods that are compared with our proposed recommender system are as follows:

- Random: this baseline method randomly samples from the list of available items at time t and recommends them to the current user who arrives at time t.

- User-centered (UC): this method has a strict attention to the preferences of the users and does not consider the preferences of the suppliers. To implement this baseline, we use an *Item-Based Collaborative Filtering* model (implemented by K-Nearest Neighbors algorithm) and predict the unknown ratings based on the past ratings of each user to the similar items. This baseline sorts the available items for a user who arrives at time *t* based on the ratings predicted by the CF model and then recommends the ones with the highest ratings to the user.

- Supplier-centered (SC): this baseline aims to satisfy the preferences of the suppliers without considering the relevance of recommendations to the users. It acquires the relevance score between the current user and the supplier of each available item using Eq. 2, sorts the available items based on their obtained relevance scores and then recommends the ones with the highest scores to the current user.

VI. EXPERIMENTAL EVALUATION

In this section, we present several of our experimental results meant to analyze the effectiveness of our proposed recommender system compared to its competitors.

A. Dataset (Table I)

Since currently there is no appropriate dataset available for the "local business-bus passengers" domain, we decided to use a combination of two well-known datasets namely MovieLens and IMDB where movies can be mapped as coupons, production companies can be used as local businesses and users can be considered as bus passengers. To do so, we merged the MovieLens dataset (100k) [14] with the IMDb movies extensive dataset (81k+) which was obtained from the Kaggle website [15]. The MovieLens dataset has 100,000 ratings from 1,000 users on 1,700 movies and the IMDb dataset has 81,274 movies produced by 30,093 production companies. Since the MovieLens data only provides the title, genre and release year of the movies, we merged the movies of these two datasets to access other metadata including duration, language, average vote and the production company that are not available in the original MovieLens dataset. We also generated a synthetic dataset containing randomly-assigned values to the preferred attributes of the production companies towards the users. These preferred attributes include age, occupation and watching status of the users. Note that all these bilateral features are either numerical or categorical. We first converted all the numerical attributes into categorical attributes (see Table I) and then transformed them all to dummy variables resulting in a total of 31 binary features for movies and 28 binary features for users.

TABLE II
DEFAULT VALUES OF THE PARAMETERS FOR EACH EXPERIMENT

number of simulations	100
number of selected users per simulation	100
maximum number of offered movies per user	5
percentage of production companies who enforce their preferences	10%
movie arrival rate (per minute)	5
user arrival rate (per minute)	1

After cleaning and merging the datasets, we ended up with 1,052 movies, 943 users, 487 production companies and 68,139 ratings.

B. Experimental Setup

In order to apply the proposed method on our dataset, we need to form the required feature vectors, which are used in the relevance functions, corresponding to the features of the movies, users and production companies.

Relevance score between a movie and a user: to implement Eq. 1, we need to build two vectors: A_i and P_u. The components of the former are the movies' attributes including genre, duration, year, language and average vote (Table I) whose values are available for each movie. The components of the latter are the users' preferred attributes towards the movies which also include genre, duration, year, language and average vote. However, since the preferences of the users towards the movies have not been explicitly specified in the MovieLens or IMDb datasets, we propose a solution to extract the *implicit* preferred attributes of each user to form the vector P_u. We assume if a movie gets a high rating (e.g. greater than 4) from a user, it means that movie has feature values that are desired by the user. Accordingly, to obtain the implicit preferred attributes of a user, first we filter out the movies that have received the highest ratings (rating score$>=4$) from the user. These movies are considered as the user's most desired movies. We then compute the mean value of each attribute of the desired movies and use them as the components of P_u. Since all attributes are already converted to dummy variables, their values are either 0 or 1. We believe that the preferred attributes of a user are the ones whose values are equal to 1 so the mean values of the preferred attributes are bigger than the mean values of other attributes. Thus, P_u represents what attributes of movies are more important or preferable by the user u. After forming A_i for each movie and P_u for each user, Eq. 1 is applied to compute the relevance between the two so-called feature vectors.

Relevance score between a user and a movie's production company: to implement Eq. 2, we compute the cosine similarity between the preferred attributes of each production company P_{s_i} (randomly generated) and the self-descriptive attributes of each user A_u. The components of these vectors are age group, occupation and watching status. The first two components are available as the users' attributes in the MovieLens dataset but the last component is generated by us. If a user has a past rating to a movie then his/her watching status is set to "seen", otherwise his/her watching status is set to "unseen". When a user is offered a movie ,which has never given a rating to, and accepts it, his/her watching status changes to "seen". After forming A_u and P_{s_i}, Eq. 2 is applied and computes the relevance score. Having the two relevance scores for each movie and each user, we can obtain the two-sided relevance score using Eq. 3.

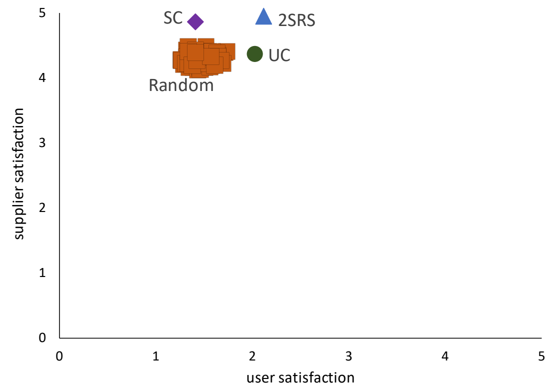

Fig. 1. User Satisfaction vs. Supplier Satisfaction with default parameters

C. Experiments and Results

In this section, we present the results of our experiments and compare our proposed method with the baselines in terms of user and supplier satisfaction. The experiments were performed on a modern computer with a Intel i5 2.3 GHz processor and 8 GB RAM. See the default values of the parameters for each experiment in Table II. As mentioned earlier, the values of the preferred attributes of the production companies towards the users (aka P_{s_i}) are generated randomly and added to the dataset. However, in a real-world scenario, less than 10% of businesses would want to enforce their preferences towards the customers and other businesses ignore their preferences and offer their promotions to everyone. For example, some of the local businesses may offer promotions to specific groups such as students or senior citizens while other businesses do not have any preference over a specific group. To this end, we randomly (using *Bernoulli Distribution*) select a certain percentage (e.g. 10%) of the production companies per preference, as the ones who want to enforce that preference and mark the rest as the ones who are happy with any customer whose preferences match their items. We tried this random selection several times to make sure that the final outcomes do not get affected by changing the data. We present our results in the following subsections.

*1) **User Satisfaction vs. Supplier Satisfaction with default parameter values (Figure 1):**:* in this experiment, we compare our proposed method, 2SRS, with the baselines in terms of user satisfaction and supplier satisfaction. We assume only 10% of suppliers would like to enforce their preferences towards the users and each user is recommended at most 5 offers (depends on how many items are available). We

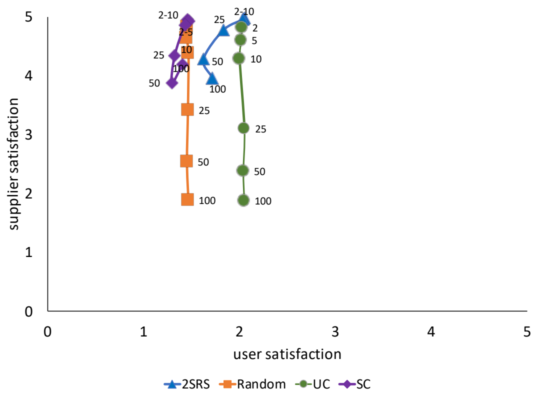

Fig. 2. Sensitivity analysis based on percentage of suppliers who enforce their preferences.

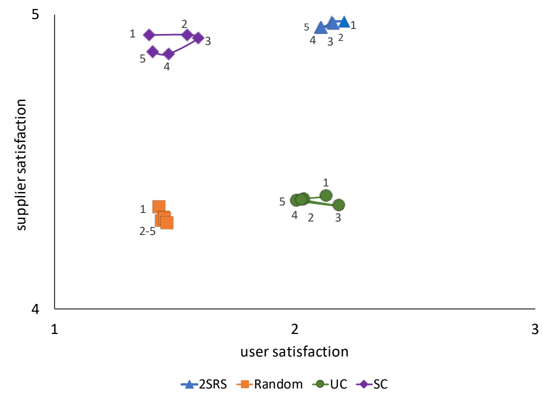

Fig. 3. Sensitivity analysis based on the number of offers to each user.

also set other parameters namely the user arrival rate and the item arrival rate to their default values which are 1 user and 5 items per minute (Table II). The mean of the satisfaction scores for both users and suppliers are computed for each simulation for each method. As you can see in Figure 1, 2SRS outperforms the baseline methods and achieves a better trade-off between the user satisfaction and the supplier satisfaction. Although the user satisfaction achieved by 2SRS is only 4% higher than the user satisfaction obtained by UC, 2SRS makes the suppliers 13% more satisfied than UC. As indicated, UC shows higher user satisfaction in comparison with SC and Random baseline because it cares more about the preferences of the users. On the other hand, SC provides higher supplier satisfaction compared to UC and Random baseline because it prioritizes the preferences of suppliers over users. Furthermore, we illustrated all the satisfaction scores (for 100 simulations) for the Random baseline instead of the mean of the values to show that this baseline never outperforms our proposed recommender system.

*2) **Sensitivity Analysis (Figures 2-5):**:* in this set of experiments, we change the configuration parameters, including the percentage of suppliers who want to enforce their preferences, the number of offers to each user, the item arrival rate and the user arrival rate, to see how the satisfaction scores are affected. We try different values for these parameters and the results show that our proposed method achieves the best trade-off between the user satisfaction and supplier satisfaction compared to the baselines when the percentage of the suppliers who enforce their preferences is less than or equal to 10%, the number of offers to each user is less than or equal to 5, the item arrival rate is greater than or equal to 5 and the user arrival rate is less than or equal to 5. It should be noted that to have more clarity in figures 3-5, we changed the scale of x and y axes from 0-5 to 1-3 and 4-5 respectively. The experiments are as follows:

1. percentage of suppliers who enforce their preferences (Figure 2): we change the percentage of suppliers who enforce their preferences by randomly selecting x% of suppliers where x = 2%, 5%, 10%, 25%, 50% and 100%, recompute the two-sided relevance scores and run 100 simulations for each new configuration. The other parameters including the number of offers, the coupon arrival rate and the user arrival rate remain at their default values and they are equal to 5, 5 and 1 respectively. As shown in Figure 2, supplier satisfaction decreases when the value of x increases. Moreover, 2SRS provides higher supplier satisfaction for all values of x compared to other baselines. There is only one exception that happens when the percentage is 100 and supplier satisfaction for SC is about 5% better than 2SRS. However, 2SRS delivers nearly 23% more user satisfaction than SC. Figure 2 also shows that user satisfaction by 2SRS is always better than Random and SC baselines but as one could see, the user satisfaction by 2SRS becomes lower than the user satisfaction by UC for x = 25%, 50% and 100%. However, the noticeable excellency in supplier satisfaction of 2SRS compared with UC (43%, 57% and 72%) for the same values of x, compensates the insignificant reduction in their user satisfaction. Plus as already stated, the percentage of the suppliers who want to apply their preferences is usually less than 10% in real-world applications.

2. number of offers to each user (Figure 3): in this evaluation, we change the number of offers from 1 to 5 and keep the other parameters as default. As illustrated in Figure 3, user satisfaction decreases (about 4% for 2SRS, UC and SC and about 2% for Random baseline) while supplier satisfaction remains almost the same (about 0.5% decrease) when the number of offers increases from 1 to 5. However, 2SRS still outperforms all baselines (for different number of offers) up to 45% in terms of user satisfaction and up to 15% in terms of supplier satisfaction.

3. item arrival rate (Figure 4): in this sensitivity analysis, the item arrival rate is changed from 1 to 5 and the other parameters remain constant. As shown in Figure 4, when the items arrive with a higher rate, the user satisfaction increases from 0.6% to 36% and the supplier satisfaction increases from 0.4% to 9%. We think this happens because when more items become available at the time when a user arrives, the

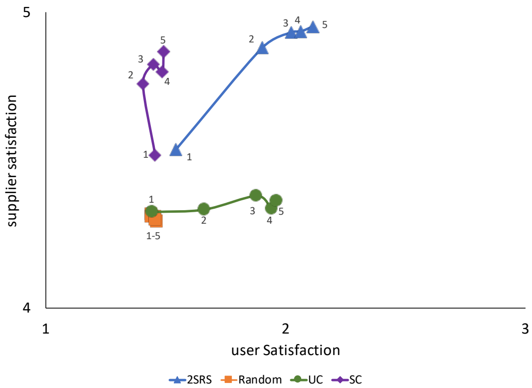

Fig. 4. Sensitivity analysis based on the item arrival rate.

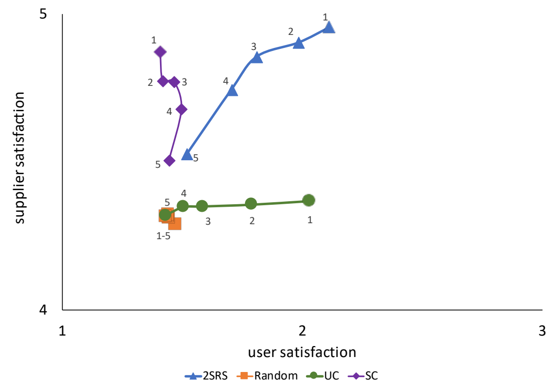

Fig. 5. Sensitivity analysis based on the user arrival rate.

probability of finding more relevant offers will increase. Our proposed method provides higher satisfaction for users and suppliers up to 36% and 15% accordingly compared to the baseline methods.

4. user arrival rate (Figure 5): in this study, we try different user arrival rates (from 1 to 5) while the other parameters remain unchanged. Figure 5 displays the performance of the methods in terms of user and supplier satisfaction when the user arrival rate changes. User satisfaction decreases from 2% to 30% and supplier satisfaction decreases from 1% to 9% when the user arrival rate increases from 1 to 5. We believe when the number of items is constant but the number of users increases, fewer relevant items will be offered to each user so the satisfaction will be reduced. So it is important that there is always a balance between the user and item arrival rates to avoid item starvation and keep the users satisfied. In addition, as one can see, SC shows a different behavior compared to other methods and delivers about 6% increase in user satisfaction when the user arrival rate increases. However, 2SRS still performs better than the other baselines (including SC) up to 40% and 14% in terms of user satisfaction and supplier satisfaction correspondingly. As may be noted, the Random baseline does not deliver much different satisfaction when the user/item arrival rate or the number of offers change.

VII. Conclusions

In this paper, we proposed a two-sided recommendation method called 2SRS to connect local businesses to passerby. This method considers the preferences of both parties and recommends the top relevant available items to each user as soon as they arrive. Our experimental results showed that our method achieves a better trade-off between the user satisfaction and supplier satisfaction (with different settings) compared to the baselines. Although our method was motivated by the "local business-bus passengers" application, we are confident that it can be employed in other applications where two stakeholders are involved and their preferences towards each other need to be fulfilled.

Acknowledgment

This work is part of the PittSmartLiving project which is supported by NSF award CNS-1739413.

References

[1] U. of Pittsburgh, "Pitt smart living project," 2016. [Online]. Available: https://pittsmartliving.org/

[2] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.

[3] P. Lops, M. De Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender systems handbook*. Springer, 2011, pp. 73–105.

[4] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The adaptive web*. Springer, 2007, pp. 291–324.

[5] R. Burke, "Hybrid recommender systems: Survey and experiments," *User modeling and user-adapted interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[6] H. Abdollahpouri, G. Adomavicius, R. Burke, I. Guy, D. Jannach, T. Kamishima, J. Krasnodebski, and L. Pizzato, "Multistakeholder recommendation: Survey and research directions," *User Modeling and User-Adapted Interaction*, vol. 30, no. 1, pp. 127–158, 2020.

[7] A. E. Roth and M. Sotomayor, "Two-sided matching," *Handbook of game theory with economic applications*, vol. 1, pp. 485–541, 1992.

[8] L. Pizzato, T. Rej, T. Chung, I. Koprinska, and J. Kay, "Recon: a reciprocal recommender for online dating," in *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pp. 207–214.

[9] J. Malinowski, T. Keim, O. Wendt, and T. Weitzel, "Matching people and jobs: A bilateral recommendation approach," in *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, vol. 6. IEEE, 2006, pp. 137c–137c.

[10] H. Yu, C. Liu, and F. Zhang, "Reciprocal recommendation algorithm for the field of recruitment," *Journal of Information & Computational Science*, vol. 8, no. 16, pp. 4061–4068, 2011.

[11] G. K. Patro, A. Biswas, N. Ganguly, K. P. Gummadi, and A. Chakraborty, "Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms," in *Proceedings of The Web Conference 2020*, 2020, pp. 1194–1204.

[12] "Poisson point process," November 2020. [Online]. Available: https://en.wikipedia.org/wiki/Poisson_point_process

[13] D. E. Knuth, *The art of computer programming*. Pearson Education, 1997, vol. 3.

[14] "Movielens," 1998. [Online]. Available: https://grouplens.org/datasets/movielens/

[15] S. Leone, "Imdb movies extensive dataset," 2019. [Online]. Available: https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset