Peer-to-Peer Match Making in Sharing Economy

From a Timebanking Approach

Ian Liu

05/01/2020

Abstract

As more innovative business types of sharing economy are being developed than ever, the problem of match making for transactions has also become necessary and essential to be studied on. In this project, we take a timebanking approach to study the match-making process occurring in a scenario of peer to peer service exchange. An algorithm that calculates the scores of similarity, complementarity, and proficiency between two agents is developed. A synthetic dataset is generated for simulating the matching process. By utilizing the scores between different profiles of agents, a recommender system is studied and presented in order to let agents/users have more choices in terms of selecting a potential candidate to exchange service.

Key Words: Sharing economy, timebanking, data mining, recommender system, decision making.

1. Introduction

As the concept of sharing economy has been globally proved to be efficient and economic, a relative new economic model has been invented and introduced, which utilizes time as a tradable currency, to let participants perform specific skills on tasks in order to exchange hours of work with other participants who request the services. On the other hand, a participant can be in both roles of service provider and requester, depending on the dynamic needs of theirs and others. During the entire transaction, although the physical money is replaced by working hours, both parties of the transaction still has achieved their objectives of exchanging services or needs by sharing common interests, or complementarities. This new type of economic model is called time-banking (Collom and Lasker (2016)).

For example, Jack will be out of town over the weekend for a conference and needs someone reliable and trustable to take care of his child. Maria, Jack's neighbor, is a professional registered nurse at a local children's hospital and will be taking a break from work over the weekend. However, due to the nature of Maria's work, she must be on night shifts 4 days a week, which makes her hardly find a proper time window to mow her lawn. However, Jack will be getting a 2-day break from work after he returns back from the conference, and he is willing to mow the lawn for Maria as a return, in exchange of Maria's time for taking care of his child over the weekend.

There actually have been a few successful platforms out there that can help the participants of timebanking set up the matches. However, things just become more complicated as the number of participants is getting larger, as well as the demands are getting more vary, which means that each participant must make a decision when multiple matches have become available for them to confront. And due to the lack of sufficient information provided to the service requester, making a decision among many choices and spending lots of time to pick potential service providers could be the major causes that the concept of timebanking haven't been adopted by majority of the public. Let's get back to Jack and Maria's case. What if there are multiple help requests that Maria could accept for the weekend? What if there are multiple babysitters available in the neighborhood that Jack gets to choose? Here comes a decision-making problem, for the system itself.

However, due to the limitations of not being able to collect enough data from real lives, we have to use synthesized data that simulates the service transaction process. So, in this project, we assume the users who are willing to use the timebanking service exchange platform have all setup their profiles, which contain the **static information** as following:

- Name
- Age
- Gender
- Location
- Education Background
- Jobs that are good at, and not good at (evaluated by numeric values from 1 to 10)
- Preference on age of the person exchanging service with
- Preference on gender of the person exchanging service with, and
- Preference on educational background of the person exchanging service with

In addition, we also assume the service requests have been made available as well, which contain **dynamic information** as following:

- Location that the service is occurring
- Exact date that the service is occurring
- Exact time that the service is occurring
- Job category that the service falls in

A sample of the synthesized data is shown in Table 1. Due to the limitation of time, the dataset is not fully and completely generated. However, this project will be further continued during the Summer 2020.

2. Mathematical Formulation

The core idea of the model follows a combination of scores of similarity, complementarity, and skill proficiency between service requester $i \in I$, and service provider $j \in J$ (Jung et al. (2016)):

$$Final\ Score_{i,j} = Score\ of\ Similarity_{i,j} + Score\ of\ Complemantarity_{i,j} + Score\ of\ Proficiency_{i\ or\ j}$$

The *Final Scores* will be later used to measure the distances between agents. This is also the foundation of the recommender system we've mentioned previously, which agents with similar distances in terms of $final\ scores$ will be recommended to each other.

2.1 Mathematical Notations

As mentioned above, there are two types of information that will be used for further analysis and calculation, **static**, and **dynamic**. In order to collect the two types of information/data, the following information will be asked and collected: * General and personal information (*Age*, *Gender*, *Educational_Background*, and *Exact Location*). * What type(s) of job(s) are you good at or do you need help with? And rank them from 1 to 10 ("10" here represents "Professionally Good at", and "1" here represents "Have No Idea How To Do It"). * Please state what preferences do you have for the potential candidates? (Age range, gender, available time windows, educational background etc.)

2.1.1 Following sets contain the Static Information of agents:

- Age of an agent is categorized into 6 groups, denoted by:
  $a \in A = \{21-30, 31-40, 41-50, 51-60, 61-70, 71+\}$

- Gender of an agent, denoted by:
  $g \in G = \{Male, Female, Not\ Specified\}$
- Educational background of an agent, denoted by:
  $e \in E = \{High\ School, Associate's, Bachelor's, Master's, Doctorate, Post - Doc, MD, JD, Other\}$
- Category of the job with proficiency levels (0-10), denoted by:
  $c \in C = \{Gardening, Housing, Cleaning, Repairing, Cooking, Gardening, Groceries, Delivery,\ etc\}$

2.1.2 Following sets contain the Dynamic Information of requests:

- Specific time of the service, denoted by:
  $t \in T = \{Time\ Slots\ of\ All\ Day\ on\ a\ 30\ min\ Inverval : 00:00-00:30, 00:30-01:00\ etc.\}$
- Specific date of the service, denoted by:
  $d \in D = \{365\ days\ in\ a\ year,\ in\ format\ of\ "MM-DD-YYYY"\}$
- Location of the agent/request, denoted by:
  $l \in L = \{All\ States\ and\ Territories\ of\ the\ U.S\ by\ Counties\}$
  - When processing the case of ride-sharing, coordinates will be used here.
- Category of the job, denoted by:
  $c \in C = \{Gardening, Housing, Cleaning, Repairing, Cooking, Gardening, Groceries, Delivery,\ etc\}$

2.2.1 Profiles of people who request services (i^{th} agent):

$$Prof_i = [Prof_{i,a}, Prof_{i,g}, ...Prof_{i,\theta}]$$

Their preferences to potential agent j are denoted:

$$Pref_i = [Pref_{i,a}, Pref_{i,g}, Pref_{i,c}......Pref_{i,\theta}]$$

And their requests are denoted:

$$Req_i = [Req_{i,l}, Req_{i,t}......Req_{i,\gamma}]$$

Where

$$\theta \in \Theta = \{A, G, E, C\}\ and\ \gamma \in \Gamma = \{L, T, D, C\}$$

2.2.2 Profiles of people who offer services (j^{th} agent):

$$Prof_j = [Prof_{j,a}, Prof_{j,g}, ...Prof_{j,\theta}]$$

Their preferences to potential agent i are denoted:

$$Pref_j = [Pref_{j,a}, Pref_{j,g}, Pref_{j,c}......Pref_{j,\theta}]$$

And their requests are denoted:

$$Req_j = [Req_{j,l}, Req_{j,t}......Req_{j,\gamma}]$$

Where

$$\theta \in \Theta = \{A, G, E, C\}\ and\ \gamma \in \Gamma = \{L, T, D, C\}$$

2.3.1 Score of similarity for profiles and preferences

- Score of similarity when two profiles are matched in terms of the general profile information (**Static**):

$$\sum_{k=1}(1_{(Prof_{i,k}=Prof_{j,k})})$$

- Score of similarity when two profiles are matched in terms of agents' preferences (**Static**):

$$\sum_{k=1}(1_{(Pref_{i,k}=Pref_{j,k})})$$

- Score of similarity when **dynamic info** from requests matches the **static info** in profiles

$$\sum_{k=1}(1_{Prof_{i,k}=Req_{j,k}})$$

- Score of similarity when **dynamic info** from requests matches the **static info** in preferences

$$\sum_{k=1}(1_{Pref_{i,k}=Req_{j,k}})$$

Where

$\sum_{k=1}(1_A)$ is an indicator function, index k denotes the elements inside of set Γ *and* $\Theta, i \neq j$.

By summating the scores similarities of all the sub-parts, the final score of similarity for agent i and j is

$$Similarity = \sum_{k=1}(1_{(Prof_{i,k}=Prof_{j,k})}) + \sum_{k=1}(1_{(Pref_{i,k}=Pref_{j,k})}) + \sum_{k=1}(1_{Prof_{i,k}=Req_{j,k}}) + \sum_{k=1}(1_{Pref_{i,k}=Req_{j,k}})$$

2.3.2 Score of Complementarity for Profiles of Users

The score of complementarity is calculated based on the proficiency ratings given by agents when they set up profiles. 1 represents the starter level and 10 represents professional level.

$$Complementarity_{i,j} = |Prof_i(c) - Prof_j(c)|$$

For instance, $Prof_i$ states that they are not good at gardening (rated "1"); while $Prof_j$ states that their gardening skill is nearly perfect (rated "9"). As result,

$$Score\ of\ Complemantarity_{i,j} = |Prof_i(c_{gardening}) - Prof_j(c_{gardening})| = |1-9| = 8$$

In conclusion, the higher score of complementarity is, the more likely two participants are a good match in terms of their complementarity over a specific job category.

2.3.3 Score of Proficiency for Users

The Score of Proficiency measures how proficient a service provider i might be on a specific job $c \in C$, C here denotes the job category, or a pool. And it sums up all the previous evaluations (in numeric scores) of the jobs that i has completed, and has it divided it by the total number of times that i has been evaluated by service requesters. $w_i(c)$ denotes the evaluation score that i has received for one job.

$$Proficiency\ _{i\ or\ j} = \frac{\Sigma_i^{N_c} w_i(c)}{\#\ of\ times\ i\ or\ j \text{ has accepted requests } (N_c)}$$

The Decision Model - Final Score for Match Making

A potential match will take both of **static** and **dynamic** information into consideration while matching pairs. We use indicator functions for the matches, which means that if there is a match between two agents over an element, the final score will increase by 1. After the system picks all the potential candidates for possibles matches, it could be possible that each i could have multiple j's to pick, and how should we make the optimal decision? We calculate potential candidates' scores of similarity, complementarity, as well as the proficiency in order to recommend the most-matched j to i.

By calculating the final scores for all agents, as shown below, there will be a list of possible matches generated that have similar distances, and the matches with the top final scores will then be recommended to agents for them to choose.

$$Final\ Score_{i,j} = Similarity_{i,j} + Complemantarity_{i,j} + Proficiency_{i\ or\ j}$$

3. The Decision Problem

In this project, the decision-maker is the system itself. It's run by an algorithm that clusters all the users that have similar profiles, as well as approximate matched requests, preferences, and interests. Then the system will decide which two candidates can be a good match by utilizing the k-means clustering technique, so that both of the users, candidates in this case, do not have to make a choice from long lists of potential candidates.

4. User Data Profile Setup and Data Collection

The process of data collection will be based on analyzing the information that users put in their profiles. In general, information needed for initiating a profile match by using Natural Language Processing techniques is listed below in [Figure 2] and [Figure 3]

In terms of data collection, I am proposing to ask a group of students at UVa to fill out a survey that contains the questions in the following diagram. This is evaluated as the data source for this project. As the users' pool is getting more sufficient, the matching processing is expected to be more and more efficient.

Table 1: Sample Data

Name	Age	Gender	City	State	Edu_Bkgd	Job_Catgr	Pref_Age	Pref_Gender	Pref_Bkgd	Time	Location	Date	Req_Job_Catgr
Maria	45	Female	Charlottesville	VA	Master	“’cooking’,’1’”, "’caring’, ’9’", "’housing’,’2’", "’repairing’,’2’", "cleaning’,’2’", "’gardening’,’2’"	41-50	Female	Bachelor	15:30	"’Charlottesville’, ’VA’"	5/1/2020	gardening
Jack	33	Male	Charlottesville	VA	Bachelor	“’cooking’,’9’”, "’caring’, ’1’", "’housing’,’8’", "’repairing’,’8’", "cleaning’,’8’", "’gardening’,’8’"	30-40	Male	N/A	12:30	"’Charlottesville’, ’VA’"	5/3/2020	caring
Mike	55	Male	Richmond	VA	High school	“’cooking’,’3’”, "’caring’, ’6’", "’housing’,’9’", "’repairing’,’9’", "cleaning’,’9’", "’gardening’,’8’"	61-70	Male	N/A	8:00	"’Richmond’, ’VA’"	5/6/2020	cooking
Jason	19	Male	Fairfax	VA	Bachelor	“’cooking’,’7’”, "’caring’, ’5’", "’housing’,’1’", "’repairing’,’1’", "cleaning’,’2’", "’gardening’,’2’"	21-30	Female	Master	9:30	"’Fairfax’, ’VA’"	6/1/2020	housing
Emily	21	Female	Charlotte	NC	Bachelor	“’cooking’,’3’”, "’caring’, ’6’", "’housing’,’7’", "’repairing’,’5’", "cleaning’,’5’", "’gardening’,’2’"	21-30	Female	N/A	10:30	"’Charlotte’, ’NC’"	5/20/2020	gardening
Anna	30	Female	Atlanta	GA	Bachelor	“’cooking’,’5’”, "’caring’, ’5’", "’housing’,’2’", "’repairing’,’9’", "cleaning’,’7’", "’gardening’,’9’"	31-40	Female	N/A	15:30	"’Atlanta’, ’GA’"	5/21/2020	housing
Sean	23	Male	D.C.	D.C.	High school	“’cooking’,’9’”, "’caring’, ’3’", "’housing’,’3’", "’repairing’,’2’", "cleaning’,’7’", "’gardening’,’4’"	21-30	Male	JD	12:30	"’D.C.’"	5/18/2020	repairing
Ian	70	Male	Alexandria	VA	Other	“’cooking’,’1’”, "’caring’, ’5’", "’housing’,’3’", "’repairing’,’3’", "cleaning’,’7’", "’gardening’,’8’"	61-70	Male	Other	8:00	"’Alexandria’, ’VA’"	5/16/2020	cooking
Jazmin	32	Female	Charlotte	NC	JD	“’cooking’,’5’”, "’caring’, ’4’", "’housing’,’2’", "’repairing’,’7’", "cleaning’,’3’", "’gardening’,’3’"	31-40	Male	Other	9:30	"’Charlotte’, ’NC’"	5/13/2020	cleaning
Katy	45	Female	Atlanta	GA	JD	“’cooking’,’9’”, "’caring’, ’9’", "’housing’,’6’", "’repairing’,’3’", "cleaning’,’5’", "’gardening’,’1’"	41-50	Male	Bachelor	10:30	"’Atlanta’, ’GA’"	5/30/2020	gardening

5. Further Plan and Additional Thoughts

5.1 K-means Clustering

K-means is one of the most famous clustering techniques that is being used to solve problems under the environment of unsupervised learning.

In the case of making matches based on profiles' contents (static data), as well as requests (dynamic), we could define the $Final\ Score\ of\ Matching$, a composite of Score of Similarity, Score of Complementarity, and Score of Proficiency, to be a new type of metric that helps determine the distance between data points (Kim and Ahn (2008)).

Using the idea of k-means to solve Peer-to-peer matching problem is non-trivial. Here, the novelty is that we consider 'matching' as an unsupervised learning problem. In fact, any unsupervised learning algorithm can be used in this case We consider matching level as a redefined metric in feature space. In addition, based on the past transactions, we can do a metric learning to learn what is the best matching metric.

And by defining and calculating the new metric, we may be able to discover a new recommender system that clusters users by their matching scores into different groups. Then the users that have the minimum distances with others in a clustering group will be recommended for further matching selection.

5.2 Collaborative Filtering

The concept of Collaborative Filtering is widely used in recommender systems. For this project, the collaborative filtering will be a comprehensive and mature tool to use to group people with similar final scores (Herlocker, Konstan, and Riedl (2000)). However, since each user has many feature, or tags, matrix factorization must be applied in order to perform further analysis (Schafer et al. (2007)). More study will be conducted in the summer.

- Use similarity of $cos(\theta)$ to calculate the vector distance between two agents:

$$similarity = cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\Sigma_{i=1}^{n} A_i \cdot B_i}{\sqrt{\Sigma_{i=1}^{n}(A_i)^2} \cdot \sqrt{\Sigma_{i=1}^{n}(B_i)^2}}$$

5.3 Some Additional Thoughts

The set of options is defined using pre-set profiles and requests, despite complexity and diversity of each participant's profile. The system would calculate scores of similarity, complementarity, and proficiency for potential participants based on the available information provided in their profiles. The stakes of the decision vary depending on the nature of each decision. However, we wish to find the best match, or the final decision with the highest final scores, for participants to choose. On the other hand, poor decisions will definitely lead to issues such as inequality, controversy, or disputation about timing hours (Doryab et al. (2017)). Since the physical money does not get involved in the transaction, it's pretty hard for participants to intuitively evaluate the work load with working hours. Having said that, it is definitely not going to work out if Maria babysits Jack's child for the entire weekend, but Jack only mows Maria's lawn for 5 minutes. This issue will be addressed in the further study.

6. Conclusion

Due to the limitation of data generation and collection, this project is not complete in terms of the implementation phase. However, since the project will be continued during the summer time, the complete implementation and results will be presented by the end of summer 2020.

Acknowledgement

I sincerely appreciate the enlightenment, indication, and advice that Professor Small and Professor Doryab have given to me through this project.

References

Collom, Ed, and Judith N Lasker. 2016. *Equal Time, Equal Value: Community Currencies and Time Banking in the Us*. Routledge.

Doryab, Afsaneh, Victoria Bellotti, Alaaeddine Yousfi, Shuobi Wu, John M Carroll, and Anind K Dey. 2017. "If It's Convenient: Leveraging Context in Peer-to-Peer Variable Service Transaction Recommendations." *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1 (3): 1–28.

Herlocker, Jonathan L, Joseph A Konstan, and John Riedl. 2000. "Explaining Collaborative Filtering Recommendations." In *Proceedings of the 2000 Acm Conference on Computer Supported Cooperative Work*, 241–50.

Jung, Hyunggu, Victoria Bellotti, Afsaneh Doryab, Dean Leitersdorf, Jiawei Chen, Benjamin V Hanrahan, Sooyeon Lee, Dan Turner, Anind K Dey, and John M Carroll. 2016. "'MASTerful'Matchmaking in Service Transactions: Inferred Abilities, Needs and Interests Versus Activity Histories." In *Proceedings of the 2016 Chi Conference on Human Factors in Computing Systems*, 1644–55.

Kim, Kyoung-jae, and Hyunchul Ahn. 2008. "A Recommender System Using Ga K-Means Clustering in an Online Shopping Market." *Expert Systems with Applications* 34 (2): 1200–1209.

Schafer, J Ben, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. "Collaborative Filtering Recommender Systems." In *The Adaptive Web*, 291–324. Springer.