Trust Prediction From User Rating Implementation Report.

Kun Ye[1]

Abstract— **This project is inspired by "Trust prediction from user-item ratings"[1]. The objective of this project is that I intend to create eight Java programs to calculate data by using the methods from the article that written by Nikolay Korovaiko and Alex Thomo. Given features such as user Id and another user Id that in the data set, the program will calculate those data in the local sever buy using different strategies, and be able to output of all the parameters that are needed for doing trust prediction.**

ACKNOWLEDGEMENT

In the beginning, I would like to express my very great appreciation to my supervisor Dr.Thomo, for his patient guidance, enthusiastic encouragement and useful critiques of this research work. He has continually encouraged me to think about this project, and when I'm having questions, he always answered patiently. Without his support, this project would not be possible. As an undergraduate student who wants to discover more in the big data world, he definitely gives me the courage and confidence to keep studying and researching.

I. INTRODUCTION

The social network is exponentially growing in recent years. Along with the popular use of the Internet, many people have decided to become a resident on the Internet. Therefore, people's concepts of society have been greatly influenced by the social network. The social network is the main platform that connects everyone on the Internet. People can get information, buying and selling through Internet; however, all of this actives are based on trust links between users. Unlike, building the trust relationship in the real world, It is hard to know what other users through Internet . The trust relationship is essential for online trading.Therefore, predicting the user will trust another user will be an important factor that changes the online trading structure. Since the machine learning technology is also growing rapidly, it will be a good idea to use machine learning skill studying the user behavior in order to make trust prediction. This project will mainly focus on implanting the trust prediction idea from the article that written by Nikolay Korovaiko and Alex Thomo.

II. RELATED WORKS

There has been some research regarding trust prediction for social network. Three researchers have done a studying on developing and improving automatic techniques that support users in making their on-line activities[2]. Another group of researchers has done a study on trust prediction from a sociological perspective[3]. This project relates to those because it will attempt to predict the trust prediction from the social network. Their studies can bring more ideas on building a better trust prediction model.

III. PROJECT DESCRIPTION

The final deliverable for the project will be eight Java programs that allow users to interact with given dataset. The user will give the two user ID, after running a specific program, such as "f1a.Java" it will output values into text files. Those text files will contain two user ID, and the value of parameter f1a of those two users, and the trust relationship of those two users. "1" indicates user1 trusts user2, "-1" implies that user1 doesn't trust user2. Users can follow these steps to get the eight intermediate sets. Those will be used in building the training model for trust prediction.

A. Prediction and Categories

The first thing I would like to do with the data is that given the attributes listing on the dataset that I downloaded from trustlet[4]. I want to make the following explanations.

- user1 is trust user2
- user2 is trust user1
- user1 and user2 trust each other

The trust relationship is identified as following:

- 1 user1 has positive trust statement to user2
- -1 user1 has negative trust statement to user2

The the factor of determining trust statement are identified as following:

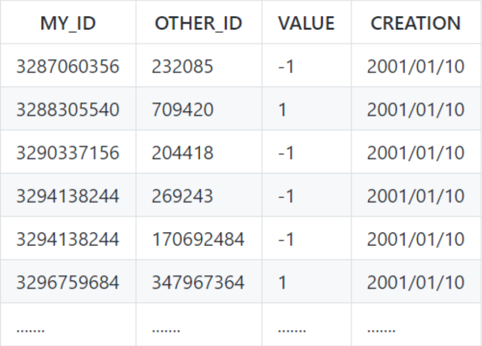
- User ID
- Item ID
- Rating
- Categories
- Reviews
- Reviews rated by other user
- Categories of reviews item
- Anonymous rating

B. Dataset

All the data comes from trustlet[4]. In this project, I mainly focus on the extended epinions dataset. In the extended epionions dataset, it contains user_rating.txt,mc.txt, and rating.txt.

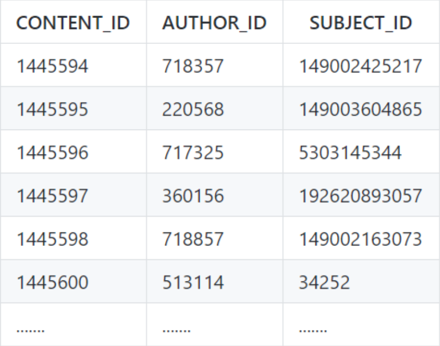
*This work was not supported by any organization

[1] Instructor: Dr.Thomo is with the Faculty of Computer Science, University of Victoria, Victoria, B.C.,Canada



MY_ID	OTHER_ID	VALUE	CREATION
3287060356	232085	-1	2001/01/10
3288305540	709420	1	2001/01/10
3290337156	204418	-1	2001/01/10
3294138244	269243	-1	2001/01/10
3294138244	170692484	-1	2001/01/10
3296759684	347967364	1	2001/01/10
....	....	....	....

Fig. 1. Information stored in user_rating.txt.



CONTENT_ID	AUTHOR_ID	SUBJECT_ID
1445594	718357	149002425217
1445595	220568	149003604865
1445596	717325	5303145344
1445597	360156	192620893057
1445598	718857	149002163073
1445600	513114	34252
....	....	....

Fig. 2. Information stored in mc.txt.

1) user_rating.txt: This text file contains the trust relationship between two users in epinion. The first row indicates that user 3287060356 distrust user 232085 on 2001/01/10[Fig 1].

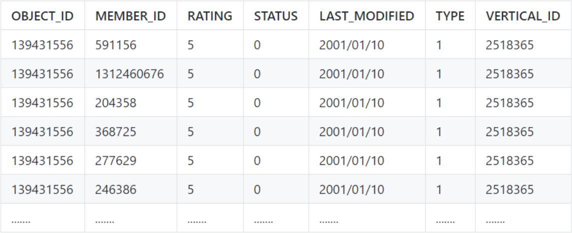
- MY_ID: This stores Id of the member who is making the trust/distrust statement.
- OTHER_ID: The other ID is the ID of the member being trusted/distrusted.
- VALUE: Value = 1 for trust and -1 for distrust.
- CREATION: It is the date on which the trust was made.

2) mc.txt: This file stores the information about the reviews, and reviewers. The first row shows the review 1445594 is written by user 718357 with subject 149002425217[Fig 2].

- CONTENT_ID:The object ID of the article.
- AUTHOR_ID:The ID of the user who wrote the article.
- SUBJECT_ID:The ID of the subject that the article is supposed to be about.

3) rating.txt: This file stores the information about the ratings and raters.The first row shows the item(review) 139431556 is rated by user 591156 with rating 5 and show the rating on 2001/01/10, and the category is 2518365[Fig 3].

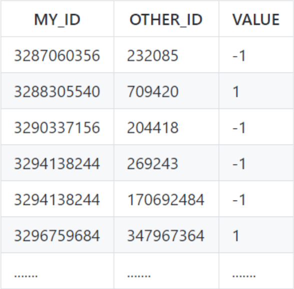
- OBJECT_ID:The object ID is the object that is being rated. The only valid objects at the present time are the content_id of the member_content table. This means that at present this table only stores the ratings on reviews and essays
- MEMBER_ID:Stores the id of the member who is rating the object



OBJECT_ID	MEMBER_ID	RATING	STATUS	LAST_MODIFIED	TYPE	VERTICAL_ID
139431556	591156	5	0	2001/01/10	1	2518365
139431556	1312460676	5	0	2001/01/10	1	2518365
139431556	204358	5	0	2001/01/10	1	2518365
139431556	368725	5	0	2001/01/10	1	2518365
139431556	277629	5	0	2001/01/10	1	2518365
139431556	246386	5	0	2001/01/10	1	2518365
....	....	....	....	....	....	....

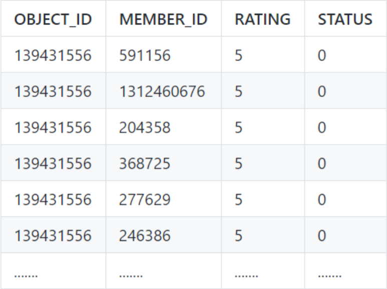
Fig. 3. Information stored in rating.txt.

- RATING:Stores the 1-5 (1- Not helpful , 2 - Somewhat Helpful, 3 - Helpful 4 - Very Helpful 5- Most Helpful) rating of the object by member.
- STATUS:The display status of the rating. 1 :- means the member has chosen not to show his rating of the object and 0 meaning the member does not mind showing his name beside the rating.
- LAST_MODIFIED:The latest date on which the member modified his rating of the object TYPE If and when we allow more than just content rating to be stored in this table, then this column would store the type of the object being rated.
- VERTICAL_ID:Vertical_id of the review.



MY_ID	OTHER_ID	VALUE
3287060356	232085	-1
3288305540	709420	1
3290337156	204418	-1
3294138244	269243	-1
3294138244	170692484	-1
3296759684	347967364	1
....	....	....

Fig. 4. user_rating.txt after cleaning



OBJECT_ID	MEMBER_ID	RATING	STATUS
139431556	591156	5	0
139431556	1312460676	5	0
139431556	204358	5	0
139431556	368725	5	0
139431556	277629	5	0
139431556	246386	5	0
....	....	....	....

Fig. 5. mc.txt after cleaning

C. Preprocessing data

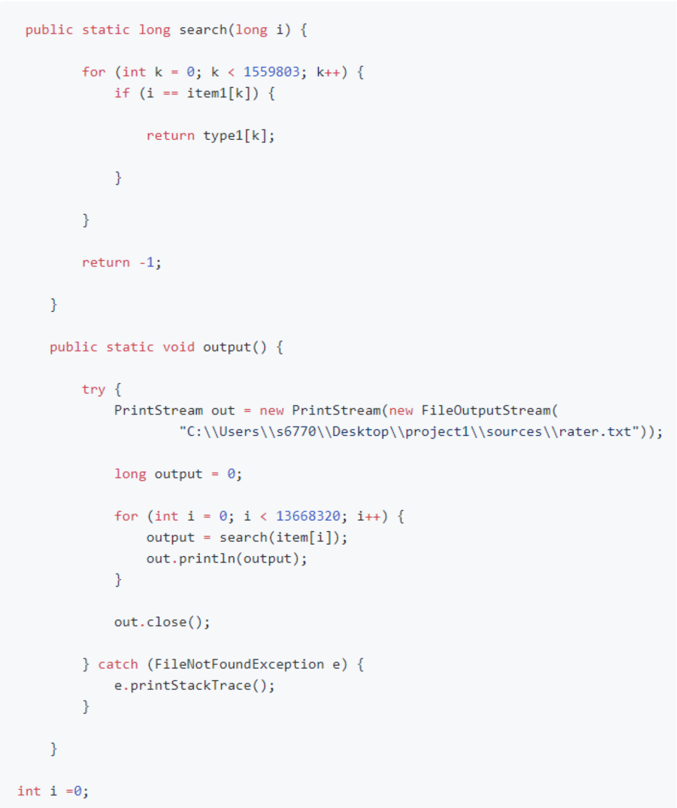
In order to improve the program's running efficiency, it is essential to use a well-processed input dataset. The original text files separate the rater, type of the item and the producer. The key idea in this processing step is to combine the rater who rated the item i, the reviewer who produced the item i, the type of the item.

1) Removing unnecessary informations: First of all, we need to clean the dataset for the better calculation accuracy and running efficiency.

- user_rating.txt: I deleted CREATION column[Fig 4].
- rating.txt: I deleted LAST_MODIFIED column and TYPE column, and VERTICAL_ID column[Fig 5].

2) Store dataset into the program: I used Scanner inputstream = new Scanner(file) to take input from the rating.txt, and mc.txt;

- item[ ]: It stores the the item id from user_rating.txt.
- user[ ]: It stores the user id from user_rating.txt.
- item1[ ]: It stores the item id from mc.txt.
- type1[ ]: It stores the type of the item from mc.txt.
- review1[ ]: It stores the reviewer id from mc.txt.

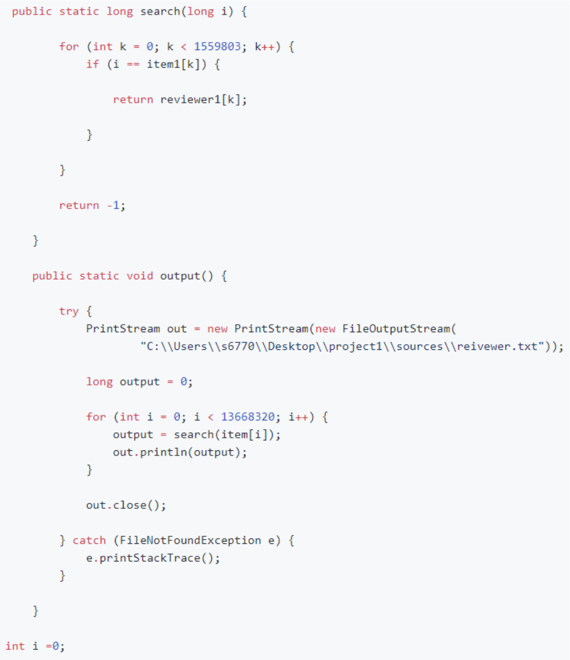


```
 public static long search(long i) {

        for (int k = 0; k < 1559803; k++) {
            if (i == item1[k]) {

                return type1[k];

            }

        }

        return -1;

    }

    public static void output() {

        try {
            PrintStream out = new PrintStream(new FileOutputStream(
                    "C:\\Users\\s6770\\Desktop\\project1\\sources\\rater.txt"));

            long output = 0;

            for (int i = 0; i < 13668320; i++) {
                output = search(item[i]);
                out.println(output);
            }

            out.close();

        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }

    }

int i =0;
```

Fig. 6. Code segment of pairing the rater with type of the item.

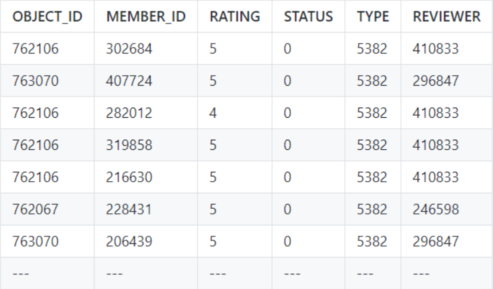
3) Pairing the rater with type of the item.: Here is code segment of paring the rater with the type of rated item[Fig 6]. The basic idea is sending every item id in item[ ] to function Search, This function will return the typed of that item. It output the type into the type.txt file.

4) Pairing the rater with reviewer.: Here is code segment of paring the rater with reviewer[Fig 7]. The basic idea is sending every item id in item[ ] to function Search, This function will return reviewer id. It output the id into the reviewer.txt file.



```
 public static long search(long i) {

        for (int k = 0; k < 1559803; k++) {
            if (i == item1[k]) {

                return reviewer1[k];

            }

        }

        return -1;

    }

    public static void output() {

        try {
            PrintStream out = new PrintStream(new FileOutputStream(
                    "C:\\Users\\s6770\\Desktop\\project1\\sources\\reivewer.txt"));

            long output = 0;

            for (int i = 0; i < 13668320; i++) {
                output = search(item[i]);
                out.println(output);
            }

            out.close();

        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }

    }

int i =0;
```

Fig. 7. code segment of paring the rater with the type of rated item



OBJECT_ID	MEMBER_ID	RATING	STATUS	TYPE	REVIEWER
762106	302684	5	0	5382	410833
763070	407724	5	0	5382	296847
762106	282012	4	0	5382	410833
762106	319858	5	0	5382	410833
762106	216630	5	0	5382	410833
762067	228431	5	0	5382	246598
763070	206439	5	0	5382	296847
---	---	---	---	---	---

Fig. 8. new rating.txt

5) Combining type.txt and reviewer.txt: After we get the two text files we can start combining rating.txt, type.txt, and reviewer.txt. Here is the new rating.txt file[Fig 8].

D. Sorting

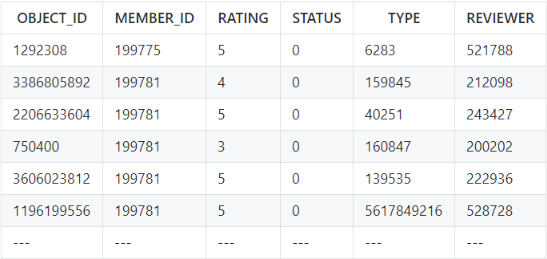
Before we start sorting, it is important to remove all the noisy data(rows with data missing).After cleaning the data, we can start sorting the rows depends on the column feature. In rating.txt, I sorted rows depends on the user id from smallest value to largest value[Fig 9].

I also create a copy of rating.txt, and I renamed it with category.txt In category.txt, I sorted rows depends on the reviewer id from smallest value to largest value[Fig 10].

In user_rating.txt, I sorted rows depend on the MY_ID[Fig 11].

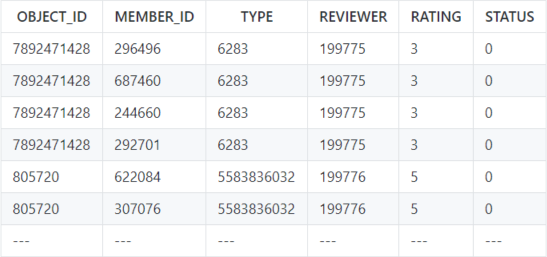
E. Java program explanation

In this project, I build eight Java classes, and each of them is corresponding with the parameter calculation in the article. All of them will take input from category.txt, rating.txt, user_rating.txt, and list.txt.



OBJECT_ID	MEMBER_ID	RATING	STATUS	TYPE	REVIEWER
1292308	199775	5	0	6283	521788
3386805892	199781	4	0	159845	212098
2206633604	199781	5	0	40251	243427
750400	199781	3	0	160847	200202
3606023812	199781	5	0	139535	222936
1196199556	199781	5	0	5617849216	528728
---	---	---	---	---	---

Fig. 9. rating.txt sorted by user id



OBJECT_ID	MEMBER_ID	TYPE	REVIEWER	RATING	STATUS
7892471428	296496	6283	199775	3	0
7892471428	687460	6283	199775	3	0
7892471428	244660	6283	199775	3	0
7892471428	292701	6283	199775	3	0
805720	622084	5583836032	199776	5	0
805720	307076	5583836032	199776	5	0
---	---	---	---	---	---

Fig. 10. rating.txt sorted by reviewer id

1) f1.Java: This java class will take input from rating.txt, user_rating.txt, and list.txt.

f1a: The key idea of this calculation is to create those functions:

- Iu: return the set of items rated by u.
- Iur: return the set of items rated r (where r 2 f1; 2; 3; 4; 5g) by u.
- ru_avg: return the average rating by user u.
- rui: return the rating of item i that rated by user u.

After creating those functions we can start putting everything together as the formula f1a in the article shows.

f1b: The key idea of this calculation is to create those functions:

- Iu_up: returns the set of item that u gives rating with 4 or with 5.
- Iu_down: returns the set of item that u gives rating with 1 or with 5.

After creating those functions we can start putting everything together as the formula f1b in the article shows. Since, f1c, f1d, f1e follow the same Strategy, I'm not going to list then below.

2) f2.Java: This java class will take input from rating.txt, user_rating.txt, and list.txt.

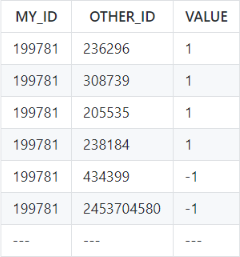
f2a: The key idea of this calculation is to create those functions:

- Cu:return the the set or multi-set (depending on the feature) of categories of the items in Iu.
- Set: set structure that stores items.

After creating those functions we can start putting everything together as the formula f2a in the article shows.

f2b: The key idea of this calculation is to create those functions:

- Cu:return the the set or multi-set (depending on the feature) of categories of the items in Iu.



MY_ID	OTHER_ID	VALUE
199781	236296	1
199781	308739	1
199781	205535	1
199781	238184	1
199781	434399	-1
199781	2453704580	-1
---	---	---

Fig. 11. user_rating.txt sorted by user one id

- Iur: return the set of items rated r (where r 2 f1; 2; 3; 4; 5g) by u

After creating those functions we can start putting everything together as the formula f2b in the article shows.

f2c: The key idea of this calculation is to create those functions:

- Cu:return the the set or multi-set (depending on the feature) of categories of the items in Iu.
- ruc: return the average rating of user u in category c.
- ru_hat return the average rating of user u to all categories.

After creating those functions we can start putting everything together as the formula f2c in the article shows.

3) f3.Java: This java class will take input from category.txt, user_rating.txt.

f3: The key idea of this calculation is to create those functions:

- return the set or multiset (depending on the feature) of categories of the reviews (items) produced by u
- Set: set structure that stores reviewer id.

After creating those functions we can start putting everything together as the formula f3 in the article shows.

4) f4.Java: This java class will take input from rating.txt, user_rating.txt, and list.txt.

f4: The key idea of this calculation is to create those functions:

- Yu: return the set or multi-set (depending on the feature) of reviewers (users) who have produced the reviews (items) in Iu
- ruy: return average rating given by a user to the reviews from a reviewer
- ru_downhat: return average rating given be a user to all reviewer.

After creating those functions we can start putting everything together as the formula f4 in the article shows.

5) f5.Java:

f5a: The key idea of this calculation is to create those functions:

- rv_avg: return average rating reviewer v received.
- ru_avg: return average rating that user u giving to the items.
- all_avg: contains the average of all rating.

After creating those functions we can start putting everything together as the formula f5a in the article shows.

f5b: The key idea of this calculation is to create those functions:

- Iu_up: return the set of item that u gives rating with 4 or with 5.
- Jv: return the set of reviews (items) produced by v.

After creating those functions we can start putting everything together as the formula f5b in the article shows.

f5c: The key idea of this calculation is to create those functions:

- Iu_dwon: return the set of item that u gives rating with 1 or with 2.
- Jv: return the set of reviews (items) produced by v.

After creating those functions we can start putting everything together as the formula f5c in the article shows.

6) f6.Java:

f6a: The key idea of this calculation is to create those functions:

- rv: return the set of all ratings for vs items.
- ru: return the set of all ratings for us items.
- Iuv: return set of vs items rated by u.
- an_rv: return the set of all anonymous ratings for vs items.
- an_ru: return the set of all anonymous ratings for us items.
- an_Iuv: return set of vs items rated anonymously by u.

After creating those functions we can start putting everything together as the formula f6a in the article shows.

f6b: The key idea of this calculation is to create those functions:

- rv_up: return the set of all ratings for vs items with rating 4 or rating 5.
- ru_up: return the set of all ratings for us items with rating 4 or rating 5.
- Iuv_up: return set of vs items rated by u with rating 4 or rating 5.
- an_rv_up: return the set of all anonymous ratings for vs items with rating 4 or rating 5.
- an_ru_up: return the set of all anonymous ratings for us items with rating 4 or rating 5.
- an_Iuv_up: return set of vs items rated anonymously by u with rating 4 or rating 5.

After creating those functions we can start putting everything together as the formula f6b in the article shows.

f6c: The key idea of this calculation is to create those functions:

- rv_down: return the set of all ratings for vs items with rating 1 or rating 2.
- ru_down: return the set of all ratings for us items with rating 1 or rating 2.
- Iuv_down: return set of vs items rated by u with rating 1 or rating 2.
- an_rv_down: return the set of all anonymous ratings for vs items with rating 1 or rating 2.
- an_ru_down: return the set of all anonymous ratings for us items with rating 1 or rating 2.
- an_Iuv_down: return set of vs items rated anonymously by u with rating 1 or rating 2.

After creating those functions we can start putting everything together as the formula f6c in the article shows.

f7.Java:

f7a:

- rv_downhat: return average rating given to the items produced by v.
- all_avg: overall average rating in the dataset.

After creating those functions we can start putting everything together as the formula f7a in the article shows.

f7b:

- ru_downhat: return average rating that u gives to the reviews
- all_avg: overall average rating in the dataset.

After creating those functions we can start putting everything together as the formula f7b in the article shows.

f7c:

- rv_downhat: return average rating given to the items produced by v.
- ru_downhat: return average rating that u gives to the reviews

After creating those functions we can start putting everything together as the formula f7c in the article shows.

7) f8.Java:

f8:

- rv_downhat: return the set of trustees of user u, i.e., those users that u trusts.
- Set: set structure that stores user id.

After creating those functions we can start putting everything together as the formula f8 in the article shows.

F. TOOLS, LANGUAGES, PLATFORMS

For version control functionality I will use Git. For preprocessing the data I will use EmEditor and Java. To implement this project I will use Java programming language. Java is a concurrent, class-based, object-oriented programing language. After considering all the details of this project. Java is a proper language that can be used to implement this project.

G. PROJECT TIMELINE

1) Studying: The studying of this project can be broken down into three main components:

- Reading the original article
- Reading the recommended system text book
- Watching the tutorial from coursera

H. Weekly Milestone

- Week of January 12: Start reading the article, and preparing for the project. Decided the high and low-level requirement of this project. Decided what software tool and programming language will be used in this project. Finding the proper dataset.
- Week of January 19: Downloaded the dataset from trustlet[2]. Combine the text file into one text file. Picking

different features from the dataset, and building charts for each feature. Following the article requirement, doing the pre-processing of the dataset.

- Week of January 26: Reading the recommender system textbook. Watching the tutorials from the course. Deciding the techniques and strategies will be implemented in this project. Made a graph chart of the structure of the project. Started building the project.
- Week of February 3: Continue working on the project. Reading the textbook. Watching the tutorial videos. preparing for the project report. Check what I missed from my program by reading the article. Fixing the bugs in my current program.
- Week of February 10: Continue working on the project. Focusing on the article. Searching necessary information for writing the project report.Trying to simplify the calculation step from the article.
- Week of February 17: Keep reading the article and searching information for writing the report. Trying to implement parameter 6 function in Python.
- Week of February 24: Finding the more proper data for this project. Complete the implementation in Python. Running the program, and it outputs the testing data.
- Week of March 4: Download the proper dataset, and starting the data preprocessing part. Implementing this project in Java. The logic will follow the logic that used in the python program. Building the input in the Java program.
- week of March 11: Completed the data preprocessing, and checking the correctness of the processed dataset. keep working on th Java program. Splitting the main program into eight individual programs that correspond to eight parameter calculations in the article.
- week of March 18: Completing the eight individual programs. Running every programs to get intermediate data set.
- week of March 25: Checking the program, and keep running them to get outputs. Comparing the new intermediate set with the old intermediate set. Writing the project report.

I. *ROLE AND ASSIGNMENT*

1) Kun Ye: He is an international student who is currently studying computer science at the University of Victoria. He is interested in Machine Learning, Neuronal network, and natural language processing. In this project, he will be responsible for finding the proper dataset. data preprocessing, implementing the project from the article, and testing the output data.

References

[1] Korovaiko, N. & Thomo, A. Soc. Netw. Anal. Min. (2013) 3: 749."https://doi.org/10.1007/s13278-013-0122-z" [On-line;accessed 18-March-2018]

[2] Piotr B, Marcin S, & Adam W (2009) Enriching Trust Prediction Model in Social Network with User Rating Similarity,"http://ieeexplore.ieee.org/document/5176100/"[On-line;accessed 16-March-2018]

[3] ang, Y., Wang, X. & Zuo, WL. J. Comput. Sci. Technol. (2015) 30: 843."https://doi.org/10.1007/s11390-015-1564-8" [On-line;accessed 18-March-2018]

[4] Extended Epinions dataset"http://www.trustlet.org/epinions.html"[On-line;accessed 19-March-2018]

IV. NOTE

A. *OUTPUTS*

Here is the sample from the intermediate set.

1) Parameter 1:

- f1a[Fig 12]
- f1b[Fig 13]
- f1c[Fig 14]
- f1d[Fig 15]
- f1e[Fig 16]

2) Parameter 2:

- f2a[Fig 17]
- f2b[Fig 18]
- f2c[Fig 19]

3) Parameter 3:

- f3[Fig 20]

4) Parameter 4:

- f4[Fig 21]

5) Parameter 5:

- f5a[Fig 22]
- f5b[Fig 23]
- f5c[Fig 24]

6) Parameter 6:

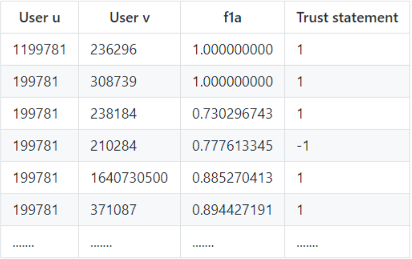
- f6a[Fig 25]
- f6b[Fig 26]
- f6c[Fig 27]

7) Parameter 7:

- f7a[Fig 28]
- f7b[Fig 29]
- f7c[Fig 30]

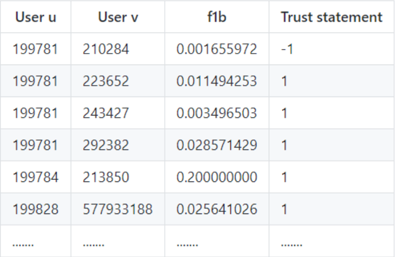
8) Parameter 8:

- f8[Fig 31]



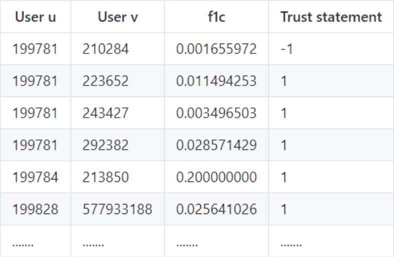
User u	User v	f1a	Trust statement
1199781	236296	1.000000000	1
199781	308739	1.000000000	1
199781	238184	0.730296743	1
199781	210284	0.777613345	-1
199781	1640730500	0.885270413	1
199781	371087	0.894427191	1
....	....	....	....

Fig. 12. f1a sample from intermediate set



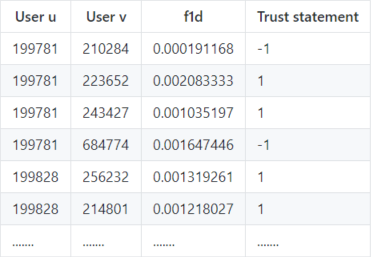
User u	User v	f1b	Trust statement
199781	210284	0.001655972	-1
199781	223652	0.011494253	1
199781	243427	0.003496503	1
199781	292382	0.028571429	1
199784	213850	0.200000000	1
199828	577933188	0.025641026	1
....	....	....	....

Fig. 13. f1b sample from intermediate set



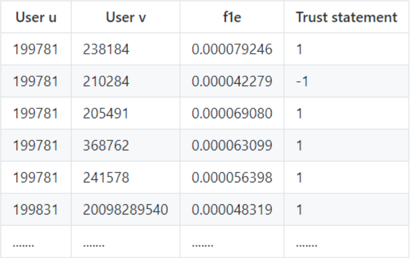
User u	User v	f1c	Trust statement
199781	210284	0.001655972	-1
199781	223652	0.011494253	1
199781	243427	0.003496503	1
199781	292382	0.028571429	1
199784	213850	0.200000000	1
199828	577933188	0.025641026	1
....	....	....	....

Fig. 14. f1c sample from intermediate set



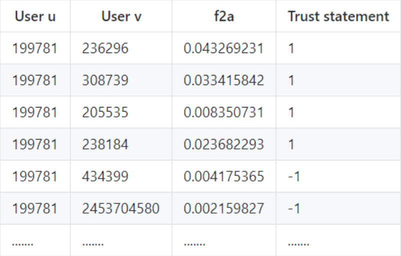
User u	User v	f1d	Trust statement
199781	210284	0.000191168	-1
199781	223652	0.002083333	1
199781	243427	0.001035197	1
199781	684774	0.001647446	-1
199828	256232	0.001319261	1
199828	214801	0.001218027	1
....	....	....	....

Fig. 15. f1d sample from intermediate set



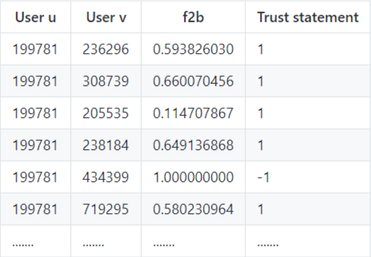
User u	User v	f1e	Trust statement
199781	238184	0.000079246	1
199781	210284	0.000042279	-1
199781	205491	0.000069080	1
199781	368762	0.000063099	1
199781	241578	0.000056398	1
199831	20098289540	0.000048319	1
....	....	....	....

Fig. 16. f1e sample from intermediate set



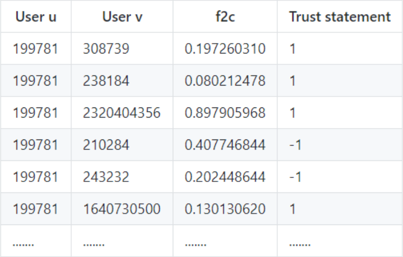
User u	User v	f2a	Trust statement
199781	236296	0.043269231	1
199781	308739	0.033415842	1
199781	205535	0.008350731	1
199781	238184	0.023682293	1
199781	434399	0.004175365	-1
199781	2453704580	0.002159827	-1
....	....	....	....

Fig. 17. f2a sample from intermediate set



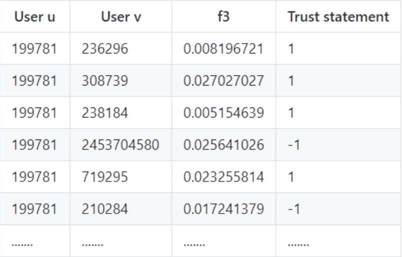
User u	User v	f2b	Trust statement
199781	236296	0.593826030	1
199781	308739	0.660070456	1
199781	205535	0.114707867	1
199781	238184	0.649136868	1
199781	434399	1.000000000	-1
199781	719295	0.580230964	1
....	....	....	....

Fig. 18. f2b sample from intermediate set



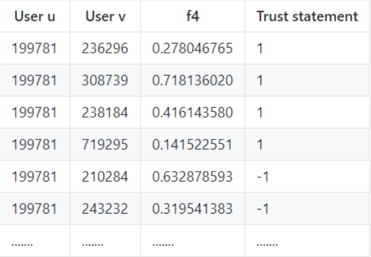
User u	User v	f2c	Trust statement
199781	308739	0.197260310	1
199781	238184	0.080212478	1
199781	2320404356	0.897905968	1
199781	210284	0.407746844	-1
199781	243232	0.202448644	-1
199781	1640730500	0.130130620	1
....	....	....	....

Fig. 19. f2c sample from intermediate set



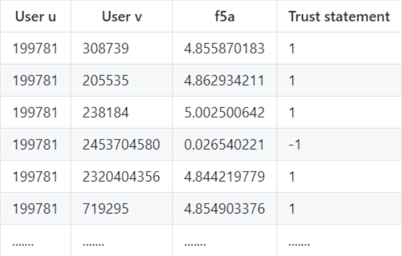
User u	User v	f3	Trust statement
199781	236296	0.008196721	1
199781	308739	0.027027027	1
199781	238184	0.005154639	1
199781	2453704580	0.025641026	-1
199781	719295	0.023255814	1
199781	210284	0.017241379	-1
....	....	....	....

Fig. 20. f3 sample from intermediate set



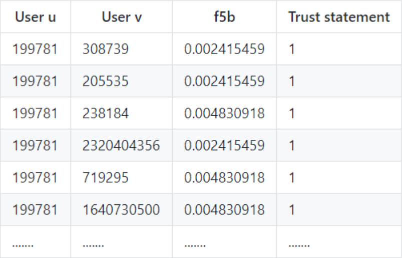
User u	User v	f4	Trust statement
199781	236296	0.278046765	1
199781	308739	0.718136020	1
199781	238184	0.416143580	1
199781	719295	0.141522551	1
199781	210284	0.632878593	-1
199781	243232	0.319541383	-1
....	....	....	....

Fig. 21. f4 sample from intermediate set



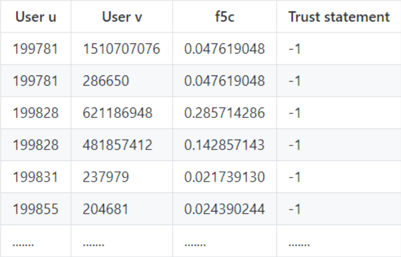
User u	User v	f5a	Trust statement
199781	308739	4.855870183	1
199781	205535	4.862934211	1
199781	238184	5.002500642	1
199781	2453704580	0.026540221	-1
199781	2320404356	4.844219779	1
199781	719295	4.854903376	1
....	....	....	....

Fig. 22. f5a sample from intermediate set



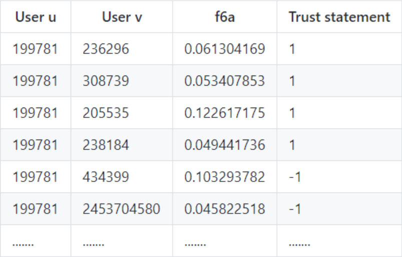
User u	User v	f5b	Trust statement
199781	308739	0.002415459	1
199781	205535	0.002415459	1
199781	238184	0.004830918	1
199781	2320404356	0.002415459	1
199781	719295	0.004830918	1
199781	1640730500	0.004830918	1
....	....	....	....

Fig. 23. f5b sample from intermediate set



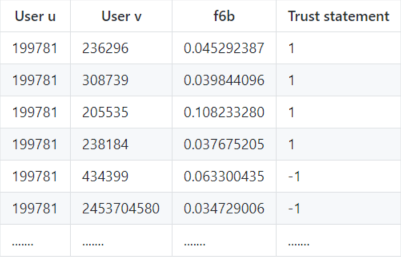
User u	User v	f5c	Trust statement
199781	1510707076	0.047619048	-1
199781	286650	0.047619048	-1
199828	621186948	0.285714286	-1
199828	481857412	0.142857143	-1
199831	237979	0.021739130	-1
199855	204681	0.024390244	-1
....	....	....	....

Fig. 24. f5c sample from intermediate set



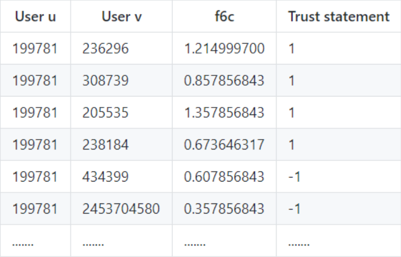
User u	User v	f6a	Trust statement
199781	236296	0.061304169	1
199781	308739	0.053407853	1
199781	205535	0.122617175	1
199781	238184	0.049441736	1
199781	434399	0.103293782	-1
199781	2453704580	0.045822518	-1
....	....	....	....

Fig. 25. f6a sample from intermediate set



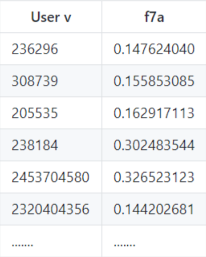
User u	User v	f6b	Trust statement
199781	236296	0.045292387	1
199781	308739	0.039844096	1
199781	205535	0.108233280	1
199781	238184	0.037675205	1
199781	434399	0.063300435	-1
199781	2453704580	0.034729006	-1
....	....	....	....

Fig. 26. f6b sample from intermediate set



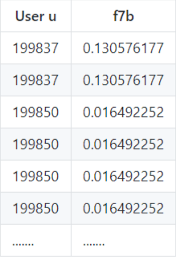
User u	User v	f6c	Trust statement
199781	236296	1.214999700	1
199781	308739	0.857856843	1
199781	205535	1.357856843	1
199781	238184	0.673646317	1
199781	434399	0.607856843	-1
199781	2453704580	0.357856843	-1
....	....	....	....

Fig. 27. f6c sample from intermediate set



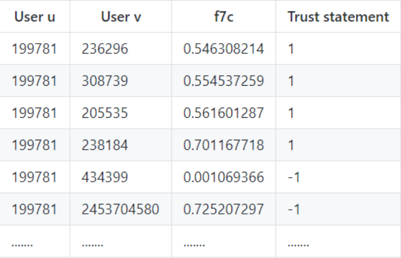
User v	f7a
236296	0.147624040
308739	0.155853085
205535	0.162917113
238184	0.302483544
2453704580	0.326523123
2320404356	0.144202681
....	....

Fig. 28. f7a sample from intermediate set



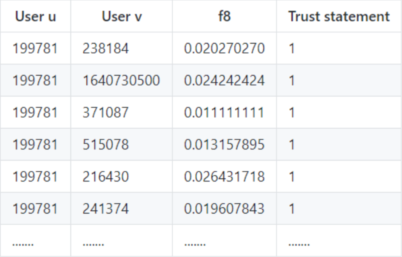
User u	f7b
199837	0.130576177
199837	0.130576177
199850	0.016492252
199850	0.016492252
199850	0.016492252
199850	0.016492252
....	....

Fig. 29. f7b sample from intermediate set



User u	User v	f7c	Trust statement
199781	236296	0.546308214	1
199781	308739	0.554537259	1
199781	205535	0.561601287	1
199781	238184	0.701167718	1
199781	434399	0.001069366	-1
199781	2453704580	0.725207297	-1
....	....	....	....

Fig. 30. f7c sample from intermediate set



User u	User v	f8	Trust statement
199781	238184	0.020270270	1
199781	1640730500	0.024242424	1
199781	371087	0.011111111	1
199781	515078	0.013157895	1
199781	216430	0.026431718	1
199781	241374	0.019607843	1
....	....	....	....

Fig. 31. f8 sample from intermediate set