The link-predicted-based friend recommender on Federated social networks

Qiurui Chen-s1988476

ABSTRACT

Federated social network is popular nowadays because it protects users' privacy. Friend recommender based on decentralized social network is important and also hard to tackle since there are only limited user information can be retrieved.

In this project, link prediction based on feature-based machine learning method is applied to implement friend recommender system. First, different sampling methods are applied to retrieve the whole social network graph, then two community detection methods, infomap and louvian algorithms, are applied to cut graph into small communities. In each cluster, different similarities, which are regarded as features, are calculated for each vertex. After feature selection, different classifiers are trained. Random forest performs best with 96% area under accuracy and 71% area under precision-recall.

1 INTRODUCTION

Socializing now seems to be an important part for Web users, and commercial interests require websites to meet the user's interest and to explore their potential needs by providing personalized services. For this purpose, the recommendation system came into being[1]. Recommender system is a system that can guide the user in a personalized way to interesting objects in a large space of possible options[2], for social network, this usually means recommendation of users. In spite of the success of major online social networking sites (like Facebook, Twitter) in attracting many users, they also raise concern about privacy issues. Especially with the recent scandal of Facebook leak 87 million user's information[3]. This problem largely comes from the fact that these social networking sites are centralized, that is to say all the user's data of one site are centrally owned by a single company, and the same entity handles all communications between its users on the site[4].

These problems motivate people considering federated social network. In a decentralized social networking framework, a user does not need to join any particular social networking service like Facebook. The users can freely choose a server to host his data just like choose an email provider. People can even set up their own server, and users from different servers can still communicate. The advantage of federated social network enables people to put as much control as possible in the hands of individual users.

There are lots of papers about recommended systems for centralized social network, like twitter. Gupta et al.[5] analyzed a few graph recommendation algorithms for twitter, these algorithms behave quite good. But the limitation of these algorithms is that they are saved in a single in-memory server, which means they have totally control of all user informations, and that is exactly the feature of centralized social network. Also this feature guarantee the quality of twitter graph, which leads these graph-based algorithms performing well. However, from the consideration of privacy, people prefer to use federated social network, which means only limited information about users can be retrieved, this is a big obstacle for recommendation system since more data can make system more accurate. The second hardship is the how to define the similarity between users and recommend based on these similarities.

Gupta et al.[5] shows that twitter use 'interest graph' to do recommend, Sarda et al.[6] come up algorithm based on trust, which contains friendship-trust and domain-expertise, to calculate the similarity score between two users. Arand et al.[7] use probabilistic matrix factorization method to calculate user interest score for each commodity. In this work, Mastodon is used to do experiment, and we will combined methods mentioned in Sarda et al.[6] and Arand et al.[7] to create a new algorithms.

The aim of this report is to recommend users for some specific users in Mastodon. Since all the users' following lists are public, different sampling methods are applied to get users' information. When the retrieved database is large enough, a graph can be generated, where each user is a vertex and a following relation is a edge. At this point, the recommendation problem can be regarded as link prediction. Feature-based machine learning method is used to do link prediction, the procedure is: calculating similarities based on different metrics and regarding these similarities as features to train a classifier[8].

2 METHODS AND MATERIALS

This part will explain how the data is retrieved and methods used, as shown in Figure 1, the main steps are:

- Prepare data
  - Removing NA: Since there are only a small amount of NA data, deleting them doesnfit influence the result.
  - Remove abnormality.
    From patient table:
    - Remove age more than 110
    - Remove date_start_of_care below than the year 2007, since ecare company started at 2007
    - Remove data that column date_end_of_care earlier than date_start_of_care

    From assessment table:
    - Remove estimated_care_minutes_weekly more than 10080 mins(7day*24h*60mins)
    - Remove duplications

    From registration table:
    - Remove care duration less than 5 mins data (registration_end_time - registration_start_time)
  - Select useful columns and combine some columns
    From registration table:
    - Financer
    - PatientID

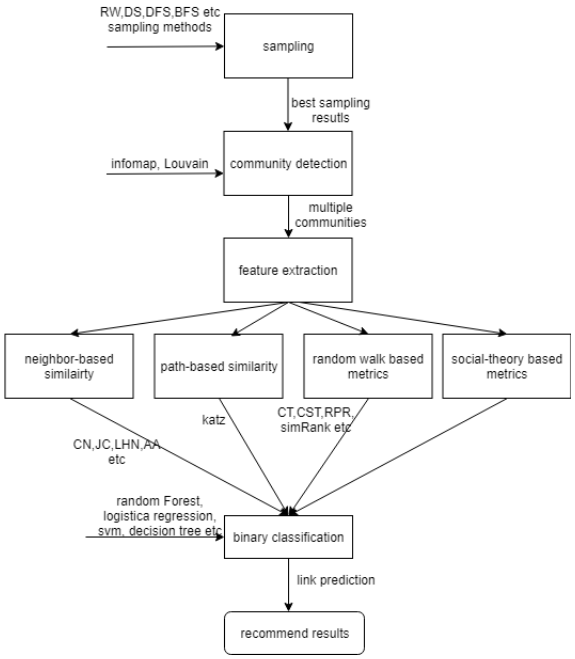


Figure 1: Overall workflow

- weekNR
- JobTitle
- TeamID
- Starttime
- Einddtime

From assessment table:

- patientID
- estimated_care_duration
- estimated_care_request
- Estimated_CareMoments_Weekly
- Estimated_Minutes_Weekly
- Advice_Instructions_Travel
- Treatments
- case_management
- Monitoring_Bevaking
- problem_environment_domain_num (sum of values in columns: income, sanitation, resodence, neighborhood/workplace safety)
- problem_psychosocial_domain_num (sum of values in columns: communication with community resources, social contact, role change, interpersonal relationship, spirituality, grief, mental health, caretaking/parenting, neglect, abuse, growth and developemnet)
- problem_physoicalical_domaintextunderscore num (sum of values in columns : Hearing, Vision, Speech and language, Oral health, Cognition, Pain, Consciousness, Skin, Neuro-musculo-skeletal function, Respiration, Circulation, Digestion-hydration, Bowel function, Urinary function, Reproductive function, Pregnancy, Postpartum, Communicable/infectious condition)
- problem_health_related_behaviors_domain_num (sum of values in columns: Nutrition, Sleep and rest patterns, Physical activity, Personal, care, Substance use, Family planning, Health care supervision,Medication regimen)

From care_plan table:

- patientID
- problemID
- signAndsymptomID
- care_plan_end_time

From signAndsympton_lookup_table

- signAndsymptonID
- problemID
- signAndSympton_name

From problem_lookup_table

- problemID
- problem_name

From patient table

- PatientID
- age
- gender
- martial_status
- living_unit

– Feature engineering.
For registration table:
  - timeDiff: the duration for each care. timeDiff = endTime - startTime
  - Mean duration for each week : mean care duration weekly for each patient. This is calculated by grouping by weekNR and PatientID column
  - Median duration for each week: median care duration weekly for each patient.
  - Total duration for each week: sum care duration weekly for each patient.

– combine data: Combined by PatientID, care_end_time in care_plan table must longer than register WeekNR. assessment_date_week_num must be the same with weekNR in registration table. Create a new comlumn named next_week_duration which week_num is one more than weekNR in registratin table. Steps are listed below:

(1) Copy and rename registration table into next_registration, delete columns except 3 columns: PatientID, total_duration_for_each_week and weekNR
(2) Rename column in next_registration from total_duration_for_each_week into duration_for_next_week
(3) Rename column in next_registration from weekNR into next_weekNR
(4) Combined next_registration and registration table by PatientId and condition: next_weekNR = weekNR +1
(5) Categorize duration_for_next_week into hours and calculate frequency for each hour categorization, combined less frequent categorizes into one. (in this experiment, we get 177 categories

corresponding to 177 hours, and we categorize hours after 15 into 5 groups since they have low frequency)

- preprocessing data
  - Normalize numerical data
  - Change categorical data into one-hot-encoding
  - Remove highly correlated features (above then .75)
  - Feature selection (based on information gain, gain ratio and symmetrical uncertainty)
- train classifier
  - Logistic regression : use L2 penalty, the regularization parameter is 0, and max iteration is 50
  - Decision tree: use gini for information gain calculation, max depth is 5 (means 5 internal node)
  - Random forest: use gini, max depth is 4, number of features to consider for splits at each node is sqrt of all features.
  - Multilayer perceptron classifier: a simple version of NN with 3 layers, input (feature) layer, middle layer (set 30 here) and output (classes) layer(20 classes here). Nodes in intermediate layers use sigmoid function and nodes in the output layer use softmax function. The logistic loss function for optimization and L-BFGS as an optimization routine is applied here.
  - One-vs-rest classifier: it cares a binary classification problem for each of the k classes.

- Collect data with 5 sampling methods and combine them.
- Detect communities.
- For each community, calculate similarities between users and regard similarities as features.
- Feature selection.
- Compare different classifiers and perform link prediction.

2.1 Notations and Definitions

- Definition 1. G = (V, E) is a network or graph where V is set of vertices and E $\subseteq$ VxV is a set of edges.
- Definition 2. A sample S is a subset of vertices, S $\subset$ V .
- Definition 3. N(S) is the neighborhood of S if N(S) = w$\in$ V-S : $\exists$ v $\in$ S s.t.(v,w)$\in$ E.

2.2 Sampling

Since the entire social network can not be derived from the federated social network, it is important to use sampling methods to get representatives.

There are lots of sampling methods, each of them is biased, for example, random work sampling method favors vertices with high degree, degree sampling is suitable for "down-sampling" a network to a connected subgraph efficiently. Also, different sampling methods prefer different network, like, 'sample edge count' sampling method is powerful for web crawling but not helpful for social network[9].

Here, static unweighted directed graph is considered to represent the social network, each vertex represents one user, and edge arrow points to following users. And limited experiment time restrict the data to be static, rather than dynamic.

Based on properties of social network graph, such as high-tailed degree distribution, small diameter[10] and high clustering coefficient, sampling methods can be measured.

In this report, five sampling methods are applied: degree sampling, random walk sampling, depth-first-search sampling, breadth-first-search sampling and randomly choose vertices. Since each method has its own downside and the goal is to get communities as much as possible, data collected from different sampling methods are combined.

Breadth-First Search (BFS). Starting with a single seed node, the BFS explores the neighbors of visited nodes. At each iteration, it traverses an unvisited neighbor of the earliest visited node[11]. In paper [12] and [13], it was empirically shown that BFS is biased towards high-degree and high-PageRank nodes.

Depth-First Search (DFS). DFS is similar to BFS, except that, at each iteration, it visits an unvisited neighbor of the most recently visited node[11].

Degree Sampling (DS). The DS strategy involves greedily selecting the node v $\in$ N(S) with the highest degree (i.e. number of neighbors), a variation of DS was analytically studied as a P2P search algorithm in [14].

Random Walk (RW). A random walk simply starts with randomly selected seed node, then performs a fixed length random walk [15].

Randomly choose vertices (RR). Randomly choose vertices is randomly selected nodes at every time, and selected fixed number nodes.

2.3 Community detection

Real-world graphs are found to exhibit a modular structure, with nodes forming groups, and possibly groups within groups. In a modular graph, the nodes form communities where groups of nodes in the same community are tighter connected to each other than to those nodes outside the community[16].

Since the entire sampling graph is quite large(nearly 20k vertices and 40k edges), calculate similarity between each non-connected vertices require powerful hardware and it's time-consuming. Based on the assumption that people are more likely connected in communities,the entire large sampling graph can be cut into several communities.

Although there are many community detection methods, but most of them are constrained to undirected network. So, here infomap algorithm and Louvain method, which are both suitable for directed graph, are implemented in this project[17].

The core of the infomap algorithm follows closely the Louvian method: neighboring nodes are joined into modules, which subsequently are joined into supermodel and so on. First, each node is assigned to its own module. Then, in the random sequential order, each node is moved to the neighboring models that results in the largest decrease of the map equation. If no move results in a decrease of the map equation ,the node stays in its original module. This procedure is repeated, each time in a new random sequential order, until no move generates a decrease of the map equation. Then the network is rebuild, with the modeled of the last level forming the nodes at this level, and ,exactly as at the previous level, the nodes are joined into modules. This hierarchical rebuilding of

the network is repeated until the map equation cannot be reduced further[18]. The map equation is:

$$L(M) = qJ(Q) + \sum_{i=1}^{m} p_i H(P_i) \tag{1}$$

$L(M)$ is the per-step description length for model partition M, q is the rate at which the index codebook is used. $H(Q)$ is the frequency-weighted average length of codewords in the index codebook. The p_i is the rate at which the module codebook i is used. $H(P_i)$ is the frequency-weighted average length of codewords in model codebook i[18]. Infomap aims at minimizing the code-length description, which means it is information theory based method.

Louvain method is a greedy optimization method that attempts to optimize the "modularity"of a partition of the network[19]. Modularity is designed to measure the strength of division of a network into modules (also called groups, clusters or communities)[20].The advantage is that it runs quite fast and easy-to implement. Although [21] says that this is the optimization method.

2.4 Similarity metrics

In this report, feature-based classification method is used to predict link. Different similarities are derived to be features. All features used in this report are based on the topological information and social theory, and they can be spitted into four categories: neighbor-based metrics, path-based metrics, random walk based metrics and social theory based metrics.

Standard notions is introduced to clarify the following description. Let lowercase letters be nodes in the social network, $\Gamma(x)$ be the set of neighbors of node x, and $|\Gamma(x)|$ be the number of neighbors of node x.

2.4.1 Neighbor-based. In a social network, people tend to create new relationships with people that are closer to them. The following approaches are based on the idea that two nodes x and y are more likely to form a link in the future if their sets of neighbors Γ(x) and Γ(y) have large overlap. Jin et al.[22] and Davidsen et al.[23] have defined abstract models for network growth using this principle, in which an edge <x, y> is more likely to form if edges <x, z> and <z, y> are already present for some z. These methods includes Common Neighbors(CN), Salton Cosine similarity(SC), Hub Promoted(HP) and Preferential Attachment(PA). Formula for each method is shown below:

Common neighbors(CN):

$$CN(x, y) = |\Gamma(x) \cap \Gamma(y)| \tag{2}$$

Salton Cosine Similarity(SC):

$$SC(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{|\Gamma(x) \cdot \Gamma(y))|}} \tag{3}$$

Hub Promoted(HP):

$$HP(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{min|\Gamma(x), \Gamma(y))|} \tag{4}$$

Preferential Attachment(PA):

$$PA(x, y) = |\Gamma(x) \cdot \Gamma(y))| \tag{5}$$

2.4.2 Path-based metrics. For this method, path between two nodes is being used for computing similarities of node pairs.

Katz: Katz[24] defines a measure that directly sums over this collection of paths, exponentially damped by length to count short paths more heavily.

$$score(x, y) := \sum_{l=1}^{\infty} \beta^l \cdot \left| paths_{x,y}^{<l>} \right| \tag{6}$$

where $paths_{x,y}^{<l>}$ is the set of all length-l paths from x to y.

2.4.3 Random walk based metrics. Social interactions between nodes in social networks can also be modeled by random walk, which uses transition probabilities from a node to its neighbors to denote the destination of a random walker from current node[25].

Hitting Time(HT): HT(x,y) is the expected number of steps required for a random walk from node x to node y.Let P=$D_{-1}AA$, where diagnoal matrix D_A of A has value $(D_A)_{i,i} = \sum_j A_{i,j}$ and $P_{i,j}$ is the probability of stepping on node j from node i. it is defined as follows [26]:

$$HT(x, y) = 1 + \sum_{w \in \Gamma(x)} P_{x,\omega} HT(\omega, y) \tag{7}$$

Commute Time(CT): Since the hitting time metric is not symmetric, commute time is used to count the expected steps both from x to y and from y to x. It can be obtained as follows[27]:

$$CT(x, y) = HT(x, y) + HT(y, x) = m(L_{x,x}^{\dagger} + L_{y,y}^{\dagger} - 2L_{x,y}^{\dagger}) \tag{8}$$

where $L^{\dagger}$ is the pseudo-inverse of matrix L = D_A− A, m is the number of edges in a social network

Cosine Similarity Time(CST): The cosine similarity time metric is based on $L^{\dagger}$ by calculating similarity of two vectors, and it can be defined as follows:

$$CST(x, y) = \frac{L_{x,y}^{\dagger}}{\sqrt{L_{x,x}^{\dagger} L_{y,y}^{\dagger}}} \tag{9}$$

simRank: Denote the similarity between objects a and b by s(x, y) ∈ [0, 1]. Write a recursive equation for s(x, y). If x = y then s(x, y) is defined to be 1. Otherwise[28],

$$s(x, y) = \frac{C}{|I(x)|\,|I(y)|} \sum_{i=1}^{|I(x)|} \sum_{j=1}^{|I(y)|} s(I_i(x), I_j(y)) \tag{10}$$

where C is a constant between 0 and 1.

Rooted PageRank(RPR): For this method, the score(x,y) is stationary distribution weight of y under the following random walk[29]:

with probability α, jump to x.

with probability 1− α , go to random neighbor of current node.

2.4.4 Social theory based metrics. In recent years, more and more works have employed classical social theories, such as community, triadic closer, string and weak ties, homophily and structure balance, to solve the social network mining and analyzing problems[25]. Liu et al.[30] proposed a link prediction model based on weak ties and three node centralities of common neighbors: degree, closeness and betweenness centrality, which performs quite

well. The model is defined as:

$$LCW(x,y) = \sum_{z} (\omega(z) \cdot f(z))^{\beta}, f(z) = \begin{cases} 1 & z \in \Gamma(x) \cap \Gamma(y) \\ 0 & otherwise \end{cases} \quad (11)$$

where $\omega(z)$ denotes the weight of node centrality, $f(z)$ is the switch function and β can adjust the contributions of each common neighbors to the connection likelihood of two nodes.

3 EVALUATION AND EXPERIMENT

3.1 Dataset and evaluation protocol

The data is collected from Mastodon.jp instance within 10/5/2018 to 15/5/2018 time-span. A ROC curve is plotting True Positive Rate (TPR) against False Positive Rate (FPR). TPR is defined as:

$$TPR = \frac{TP}{TP + FN} \quad (12)$$

FPR is defined as :

$$FPR = \frac{FP}{FP + TN} \quad (13)$$

where TP is true positive, TN stands for true negative, FP is false positive, FN represents false negative. Since classifier has no false positives, the higher the area under the ROC curve, the better the model is[31]. A precision recall curve is plotting Precision against Recall. Precision is defined as:

$$precision = \frac{TP}{TP + FP} \quad (14)$$

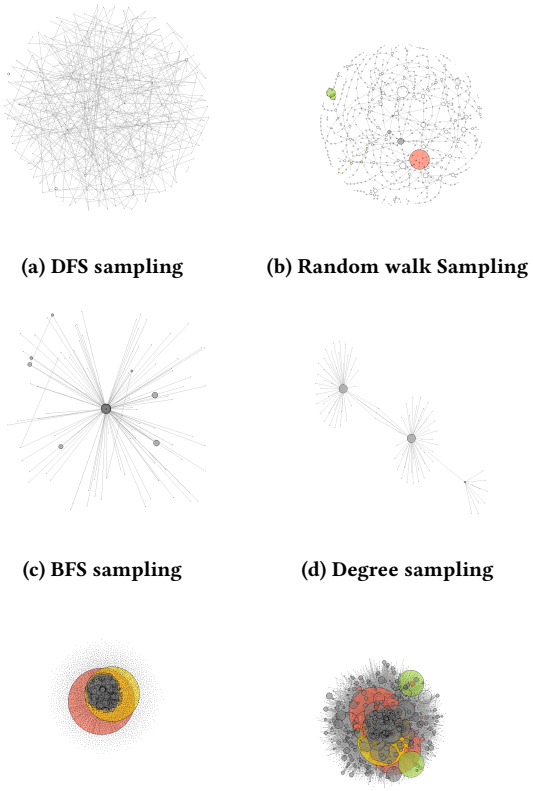
$$recall = \frac{TP}{TP + FN} \quad (15)$$

The precision recall area under curve (PR AUC) is just the area under the PR curve. The higher it is, the better the model is. Since there are many more true negatives(no link between nodes) than positives(links between nodes), the dataset is imbalance, and accuracy also include TNs, area under PR is more robust evaluation method.

3.2 Experiment

The steps of experiment is shown below:

(1) Use different sampling methods, and check how many communities can be detected for each sampling method.
(2) Combine data to get communities as much as possible since no single method performs good in terms of community numbers.
(3) Use infomap and Louvain methods to detect community for the combined graph.
(4) To simply computation, only select communities with more than 50 vertices.
(5) Save community subgraphs.
(6) Write functions that calculate the features, which are RPR, simRank, katz, CT, CST and social based similarity.
(7) Within each community, apply above functions and calculate features between each node and save the label (when there is link between two nodes, label is one, vice verse).
(8) Use 10-fold cross validation to train classifiers, which are SVM, logistic regression, decision tree and random forest.
(9) Evaluate classifiers' performance based on area under ROC curve and area under PR curve.



(a) DFS sampling **(b) Random walk Sampling**

(c) BFS sampling **(d) Degree sampling**

(e) Randomly choose vertex method sampling **(f) Final sampling graph**

Figure 2: Sampling results with different methods

3.3 Result analysis

Sampling results:

Figure 2 shows results for all sampling methods, which clarifies each sampling method properties. Different color indicate different communities, which is based on infomap algorithm, Vertices's size correspond to its out-degree.

Figure 2a and 2b indicate that DFS and RW have similar behavior. They perform quite well since multiple communities exist and also vertices are connected quite tightly, which will guarantee classifier performance.

Although RR runs fast, but figure 2e shows that it contains lots isolated vertices, which is worthless for network analysis.

Figure 2d for DS and figure 2c for BFS behave quite similar,they both explore vertices deeply.

To improve classifier performance, communities should be retrieved as many as possible. But each singe method only provides limited clusters, whereas the combination of data from all sampling methods can produce multiple communities.

Also, the combined graph should suit social network graph properties well, that is: power-law-fit, small average distance and large clustering coefficient. Figure 3 and Table 1 prove that the final graph performs well.

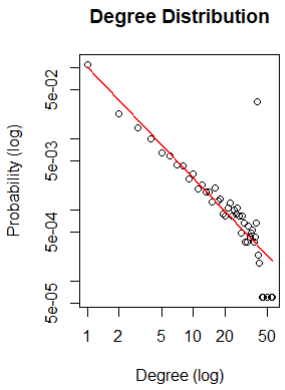


Figure 3: Power law degree distribution of final sampling network

Table 1:Properties of different sampling methods						
property	RW	RR	DFS	BFS	DS	final sampling
average distance	21.8	3.47	18.28	5.62	1.74	6.79
cluster coefficient	0.016	0.002	0	0.017	0.005	0.008
isolated vertex percentage	0.15%	28.6%	5%	0.45%	0%	22.2%

Figure 4 and 5 show the community detection based on infomap and louvian algorithm respectively, different color corresponds to different cluster. Since the network is quite big, it only display communities with more than fifty vertices. In this project, because infomap method still return a gigantic community, louvian algorithm is then applied to split this big cluster further.

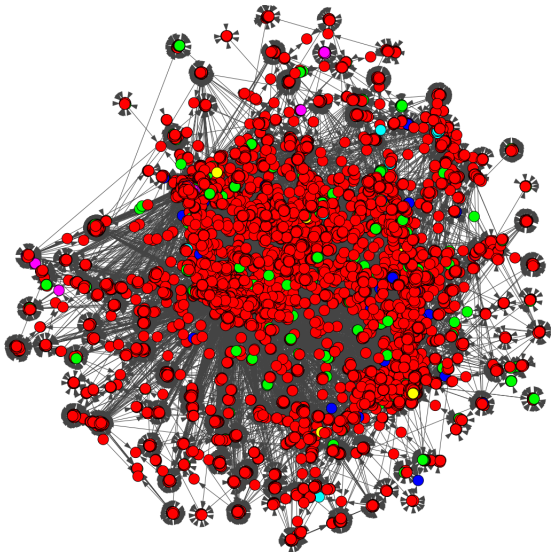


Figure 4: Community detection based on infomap algorithm

At first, 13 features are used. Figure 6a shows the correlation matrix of these features, blue indicates positive correlation, whereas red implies the negative correlation, and white color shows no relationship between each pair features, which suggests that they

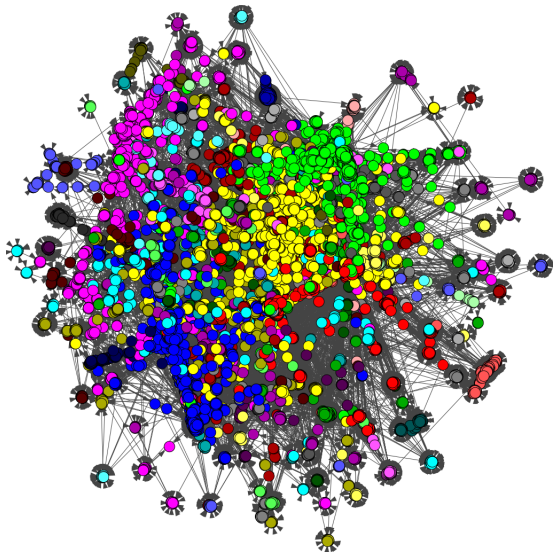
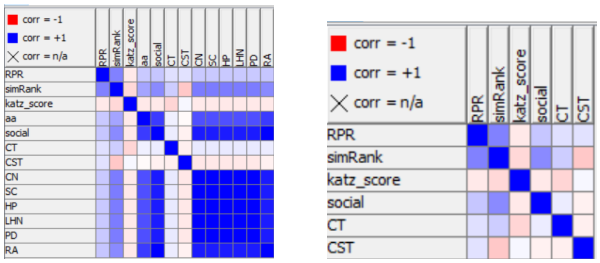


Figure 5: Community detection based on louvian algorithm



(a) Original features **(b) Final features**

Figure 6: Feature selection

are perfect features for the classifier. According to the correlation matrix, features are selected and only six features are remained: RPR, simRank, katz score, CT and social-based similarity.

After getting features, different classifiers are trained. Table 2 shows the classier performance without feature selection. Table 3 shows the classifier performance with selected features. The best result is random forest with 96% area under ROC curve(AUROC) and 71% area under Precision-Recall curve(AUPR).

Table 2:Classier performance with 13 features		
classifier	AUROC	AUPR
linear SVM	0.74	0.39
logistic regression	0.66	0.28
decision tree	0.68	0.05
random forest	0.865	0.653

Table 3:Classier performance with 6 features		
classifier	AUROC	AUPR
linear SVM	0.76	0.41
logistic regression	0.67	0.29
decision tree	0.76	0.30
random forest	0.96	0.71

4 CONCLUSIONS

For security consideration, federal social network is more preferable. And implementing a friend recommender based on limited user info is an important task to be tackled.

This report uses feature-based machine learning method to predict link and recommend friends. Since the social network graph collected from Mastodon website is huge, and based on the assumption that vertices are more likely to be connected within the same community, infomap and Louvain community detection methods are implemented to get smaller community subgraphs. And different features are collect. They are all similarity metrics used broadly these days. After feature selection, multiple classifiers are trained with 10-fold cross validation since the dataset is highly imbalanced with much more TNs, and random forest performs best with 96% AUROC and 71% AUPR.

This project is based on the assumption that vertices are more likely to be connected within the same communities, which means, links between different communities are not considered. But there is still small probabilities that users connected within different clusters.

Also, since the project is required to be done within a short time, the data-set is not time-involved. However, data set with large time span will improve evaluation to be more roust.

There are still lots of ways to improve the friend recommender performance, such as taking user profile context information into consideration to calculate similarities based on the these context.

REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*.

[2] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*.

[3] Facebook scandal 'hit 87 million users'. *BBC News*.

[4] Anthony Gentilucci Dan Stefanescu Ames Bielenberg, Lara Helm and Honggang Zhang. The growth of diaspora - a decentralized online social network in the wild. *Global Internet Symposium 2012*.

[5] Jimmy Lin Aneesh Sharma Dong Wang Pankaj Gupta, Ashish Goel and Reza Zadeh. Wtf: The who to follow service at twitter. *Twitter, Inc*.

[6] Debdoot Mukherjee Smruti Padhy Karan Sarda, Priya Gupta and Huzur Saran. A distributed trust-based recommendation system on social networks. *IEEE Workshop on Hot Topics in Web Systems and Technologies*.

[7] Jeremy Handcock Jorge Aranda, Inmar Givoni and Danny Tarlow. An online social network-based recommendation system. *Toronto, Ontario, Canada*.

[8] Nowell LD and Kleinberg J. The link prediction problem for social networks. *Proceedings of the twelfth international conference on information and knowledge management (CIKM)*, 2004.

[9] Fariba Karimi Jrgen Pfeffer JClaudia Wagner, Philipp Singer and Markus Strohmaier. Sampling from social networks with attributes. *International World Wide Web Conference Committee*.

[10] Leman Akoglu Mary McGlohon and Christos Faloutsos. Statistical properties of social networks. *Social Network Data Analytics*.

[11] R. L. Rivest T. H. Cormen, C. E. Leiserson and C. Stein. Introduction to algorithms. *McGraw-Hill Science / Engineering / Math, 2nd edition*, 2003.

[12] A. Markopoulou M. Kurant and P. Thiran. On the bias of bfs. *Arxiv e-print (arXiv:1004.1729v1)*, 2010.

[13] M. Najork. Breadth-first search crawling yields high-quality pages. *In WWW fi01*.

[14] A. R. Puniyani L. A. Adamic, R. M. Lukose and B. A. Huberman. Search in power-law networks. *Physical Review E*, 2001.

[15] L. Lovasz. Random walks on graphs: A survey. *Combinatorics: Paul Erdos is 80*, 1994.

[16] L. Akoglu M. McGlohon and C. Faloutsos. Statistical properties of social networks. *Chapter in Social Network Data Analytics, Springer Science+Business Media*.

[17] Michael J Bommarito II. Summary of community detection algorithms in igraph 0.6. *R-Bloggers*.

[18] Andrea Lancichinetti Ludvig Bohlin, Daniel Edler and Martin Rosvall. Community detection and visualization of networks with the map equation framework. *Measuring scholarly impact: theory and practice (Springer)*.

[19] Renaud Lambiotte Vincent D Blondel, Jean-Loup Guillaume and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*.

[20] M. E. J Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America*.

[21] F. D. Malliaros and M. Vazirgiannis. Clustering and community detection in directed networks: A survey. *Phys. Rep*.

[22] Michelle Girvan Emily M. Jin and M. E. J. Newman. The structure of growing social networks. (2001). 2001.

[23] Holger Ebel Jorn Davidsen and Stefan Bornholdt. Emergence of a small world from local interactions: Modeling acquaintance networks. *Physical Review Letters*, 2002.

[24] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, (18(1):39fi?!43), 1953.

[25] YuRong Wu Peng Wang, BaoWen Xu and XiaoYu Zhou. Link prediction in social networks: The state-of-the-art. *Science China Information Sciences*.

[26] Renders J M et al Fouss F, Pirotte A. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2007.

[27] Liben-Nowell D. and J. M Kleinberg. The link-prediction problem for social networks. *J. Am. Soc. Inform. Sci. Technol*, 2007.

[28] G. Jeh and J. Widom. Simrank: A measure of structural-context similarity. *In ACM SIGKDD In- ternational Conference on Knowledge Discovery and Data Mining*, 2002.

[29] V. Dave Y. Zhang H. H. Song, T. W. Cho and L. Qiu. Scalable proximity estimation and link prediction in online social networks. *The 9th ACM SIGCOMM conference on Internet measurement conference*, 2009.

[30] Haddadi H et al Liu H, Hu Z. Hidden link prediction based on node centrality and weak ties. *EPL (Europhysics Letters)*.

[31] Chioka. Differences between receiver operating characteristic auc (roc auc) and precision recall auc (pr auc).