B.Comp. Dissertation

Link Prediction in Co-authorship Network

Submitted by:

Le Nhat Minh

A0074403N

Department of Electrical & Computer Engineering

National University of Singapore

2013/2014

Project ID: H079970

In partial fulfilment of the

requirements for the Degree of

Bachelor of Computer Engineering

ABSTRACT

Link prediction is one highlighted aspect of network analysis. Many popular applications include suggesting friends on Facebook and LinkedIn are using link prediction to feature its services. As a researcher, we are particularly interested in link prediction for scholar social network. In the research area, collaborations among authors often form a network of connections which defines the co-authorship network. It is proved that collaborations between authors/researchers often yield more fruitful results than individual researchers' performance. It comes to a sense that individual researcher might find himself want to collaborate with potential researchers. Predicting the evolution of co-authorship network thus plays a significant role in (1) analyzing the trend of the structure of scientific collaborations; (2) detecting potential research communities as well as their evolutions; (3) assessing the future influence of scientists; and (4) recommending companions, assistants, or colleagues for individual researchers. Realized the importance of link prediction, we are motivated to enhance its performance especially in co-authorship network. In this work, we propose new features that help to predict the collaborations among scholars, and analyze how these features work.

Keywords: Link prediction, supervised learning, co-authorship network

ACKNOWLEDGEMENT

I would like to personally thank the following people who supported me in this project with my sincere gratitude. To my advisor Professor Min-Yen Kan and PhD students from Web Information Retrieval/ Natural Language Processing Group(WING) for their assisting and massive help during this year. I want to special thank to Dong Yuan Lu, Tao Chen and Wang Aobo, I could not finish my project without their greatest help. Lastly I would like to express my deepest gratitude to my supervisor Professor Min-Yen Kan as well as Professor NG Hwee Tou for their feedbacks, guidance and advice throughout the whole project.

TABLE OF CONTENT

LIST OF FIGURES/TABLES

I. INTRODUCTION

1.1 Problem Background

Co-authorship network reflects the collaborations among scholars, scientists and researchers. Those are people, who co-author books, journal articles, research papers, etc. Results of their academic works indicate the links among themselves within this network. The dynamic nature of collaborative links implies the co-authorship network as an exemplar of temporal and evolving social network. It grows and changes quickly over time through the addition of new edges, signifying the appearance of new interactions in the underlying social structure. A traditional definition of link prediction problem is expressed concisely by Liben-Nowell et al. [14]: ‘‘Given a snapshot of a social network at time *T*, we seek to accurately predict the edges that will be added to the network during the interval from time *T* to a given future time *T+1* ’’ as illustrated in figure 1. The same issue can be addressed to co-authorship network, i.e. provided that we have a snapshot network of researchers at a specific time, we can correctly predict collaborations among researchers in the future.

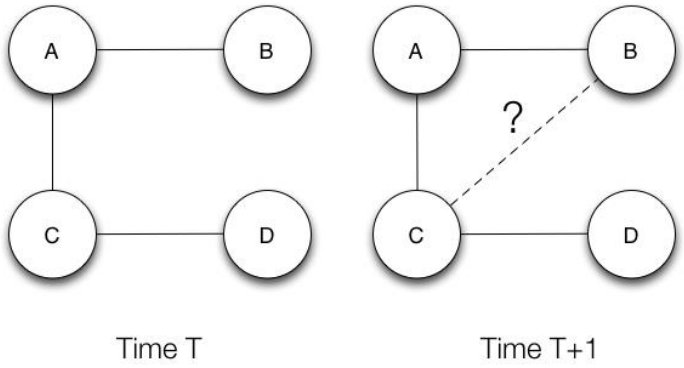


Figure 1: Traditional definition for link prediction

Many related works have proved that information about future interactions can be extracted from network topology – structure of network. The problem which we are addressing is whether we can make better prediction using features which are independent of the network structure? For example, if we take into account social features of the researchers themselves such as their institutes, their conferences attended or semantic features such as keywords and abstracts of the papers, perhaps we could improve the link prediction.

In this work, we have analysed a co-authorship network along with link prediction strategies. Our approach will use the probabilistic model and improve the performance of the model via training features. By picking up useful features, we hope to construct a supervised learning framework which incorporates all features to make link prediction accurately for co-authorship. In addition, we propose a feature that captures the productivity of author over time by observing the event of his or her co-authorship with the expectation that this new feature will improve link prediction.

1.2 Related Work

Although co-authorship network is a class of social network, there are several differences between them. Social networks like Facebook or Twitter grow more dynamically and changing more rapidly than co-authorship network. It may take days or weeks to form numerous of connections in social network but years for new co-authorships. Also social networks allow individuals to remove friends which result in changing network structure and consequently affecting the link prediction in the future. In social network, many characteristic of the node could be took into account for link prediction such as users' hobbies, relationships , hometown locations, etc. while in co-authorship network, collaborations among authors could be either cross-institute or cross-national as long as the authors share the same area of research. Lot of works have been devoted to deal with the link prediction in social network in general as well as the co-authorship network in specific such as work of Getoor et al.[7], Murata, T et al.[8], Brandao et al. [15].

Overall, there are 3 mainstream link prediction strategies: similarity based strategies, maximum likelihood algorithms and probabilistic model as noted in the survey by Lu and Zhou [9].

1.2.1 Similarity based strategies

To predict whether a link will appear or not, each pair of nodes within the network is assigned a score called similarity or proximity in literature. Intuitively, the links having higher similarity score are supposed to be of higher existence likelihoods. So far, majority of previous works in link prediction focus on the application of similarity measures such as (1) path distance, (2) common neighbours, (3) Katz clustering, (4) Jaccard, (5) PageRank , (6)

Adamic/Adar, etc. to non-connected pairs of nodes at present time in order to predict new connections in the future. The baseline for this approach was introduced by Liben-Nowell et al. [14]. New links can be predicted for instance by ranking the pairs of nodes according to their proximity scores which are mentioned above. Besides those measures, many variants of similarity measures were also introduced in attempt to improve the accuracy in link prediction. For example, Liu [1] has suggested "AuthorRank" metric which uses a normalized weight instead of the degree distribution $\frac{1}{k(A)}$, where *k(A)* denotes the outgoing links from node A for the traditional PageRank algorithm. In [11], the authors recommended another method of proximity measure called "PropFlow", which is also derived from PageRank. The "PropFlow" method suggested that the similarity between two nodes could be estimated by finding the probability that a restricted random walk starting at node A ends at node B in *n* steps or fewer using link weights as transition probabilities. As such there are plenty of hybrid methods to be considered in proximity link prediction. Obviously, the advantage of this framework is that it is simple and doesn't take so much time since it directly picks up the link among nodes in the network based on the ranking. Thus it can handle a very large network.

1.2.2 Maximum likelihood estimation

The link prediction problem can be defined as determining the probability that a link between two authors forms during the next period of time. The task of distributing probability to a non-connected link required a model in which the probability could be calculated. The model could be specified by human or recommended via machine learning.

A framework that is mentioned above is maximum likelihood estimation in which the author usually predefines some organizing principles of the network structure, with the detailed rules and specific parameters. Those principles are obtained by maximizing the likelihood of the observed structure. Then, the likelihood of any non-connected link can be calculated according to those rules and parameters. One model of this approach is "Stochastic Block Model" [10] where nodes are partitioned into groups and the probability that two nodes are connected depends solely on the groups to which they belong. Another model is "Hierarchical Structure Model" [8] which is focusing on the hierarchical organization of the network. It takes into account the *ancestors* of the nodes for link likelihood prediction. These

frameworks requires knowledge of the network structure and usually can only be applied to particular network; in those cases, "Stochastic Block Model" is useful for air-transportation network while "Hierarchical Structure Model" is designed for metabolic or brain network. Unfortunately, co-authorship network is a very sparse network with no hierarchical structure. Therefore, although the maximum likelihood methods provide very valuable insights into the network organization, this framework is not suitable to the purpose of link prediction in co-authorship network.

1.2.3 Supervised Learning

The probabilistic framework, on the other hand, will optimize a built target function to establish a model composed of a group of parameters. Given a network, we could derive multiple features from many dimensions. For example, within the links' dimension alone, there are many graph-theoretic features (node degree, common neighbours, path lengths, etc. which are also the similarity measures as mentioned above); or within the semantic' dimension, features such as key words of papers [13], institutional affiliations [15] are recommended as useful features for capturing non-existed links. Those features are then combined together to form feature vectors which are used to classified by supervised learning algorithms such as (1) Naives Bayes, (2) decision tree, (3) multiplayer perception, (4) support vector machine ,etc. Then the probability of the existence of a nonexistent link (A, B) is estimated by the conditional probability $P\ (S_{AB} = 1 |\ group\ of\ features)$, which literally means how similar a non-exist link' features are the same as the given group of features of the existent links. As noted by Lu, L. et al. [9], the supervised learning approaches seem to perform better than proximity in making link prediction, capturing effectively and efficiently the network dynamics from a time series of network snapshots. However, they still haven't fully exploited their models' ability on taking features other than path counts, which means there are other new dimensions to approach to feature a network graph.

In summary, maximum likelihood estimation required prior knowledge of network structure, which is hard to obtain for our co-authorship network, and similarity based method, in spite of its simplicity, might not be effective in predicting the collaborations among scholars. Therefore, we rely on the supervised learning framework, by utilizing the features used in the prior works, as well as our new proposals, to predict the future links in co-authorship network.

II. Method

The workflow of our supervised learning framework is summarized as below and its graphical representation is showed in Figure 2.

Step 1: Build up the network from the dataset obtained.

Step 2: Extract features to denote each link.

Step 3: Split the dataset into two sets: training set and testing set.-Train predictors on the training set and evaluate their performances on the testing set.

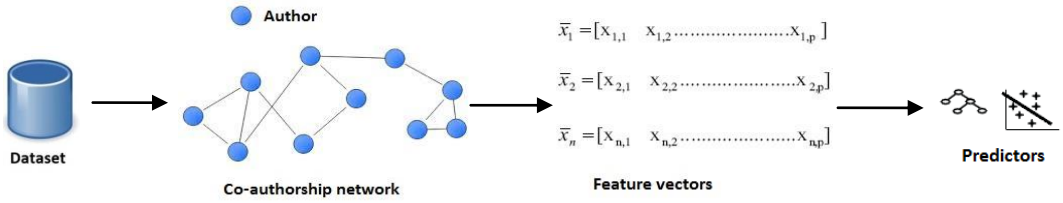


Figure 2: Building link prediction from network structure

In supervised learning, features play a critical role in learner's performance; therefore, the focus of our project is to explore effective features for predicting co-authorships. In the following, we first introduce the dataset used in our study, as it is highly related to the information we could obtain, and then we discuss the features used in the prior works, and our proposed features.

Attribute name	*Formula*
Sum of Paper	$paper(i) + paper(j)$
Shortest Path (Shortest Distance)	$\min\{s \mid path_{ij}^{s}\} > 0$
Address of University	$\frac{\lvert Address(v_i) \cap Address(v_j) \rvert}{\lvert Address(v_i) \cup Address(v_j) \rvert}$
Common Third Neighbor	$\lvert \tau_3(v_i) \cap \tau_3(v_j) \rvert$
Affiliation	$\begin{cases} 1, & if\ same\ affiliation \\ 0, & otherwise \end{cases}$
PageRank	$(1-d) + d \sum_{v \epsilon \Gamma(a)} \frac{P(v)}{\lvert \Gamma(v) \rvert}$
Preferential Attachment	$\lvert \tau_3(v_i) \rvert . \lvert \tau_3(v_j) \rvert$
Adamic Adar	$\sum_{v_h \in \tau(v_i) \cap \tau(v_j)} \frac{1}{\log \lvert \tau(v_k) \rvert}$
Keyword	$\frac{\lvert Keyword(v_i) \cap Keyword(v_j) \rvert}{\lvert Keyword(v_i) \cup Keyword(v_j) \rvert}$
Conference	$\frac{\lvert Conference(v_i) \cap Conference(v_j) \rvert}{\lvert Conference(v_i) \cup Conference(v_j) \rvert}$
Productivity	$\sum_{k=0}^{n} \frac{[n(k+1) - n(k)]^{m(K+1)-m(k)}}{T(k+1) - T(k)}$

Table 1: Attributes computed for each pair of node (v_i, v_j).

2.1 Dataset

The data of authors is retrieved from ArnetMiner (http://arnetminer.org/). ArnetMiner is a digital library which aims to provide comprehensive search and mining services such as Expert Search, Publication Search and Conference Search. Its API is strongly helpful for crawling data since we can specify our searching category in the URL with particular parameters. For example, searching for an author profile data can be done with the following URL: arnetminer.org/services/search-expert?q=[q]&u=[u]&start=[start]&num=[num]&outputjs=[outputjs]&varname=[varname,in which u is for the username, q is for keyword, etc. More information can be found at the website http://arnetminer.org/RESTful_service#b723.

Sample Output


```
{
    "time_elapsed":153.0,
    "total_result_count":8804,
    "results":
    [
        {
            "id":1158861,
            "name":"M. Irani"
        },
        {
            "id":123223,
            "name":"Andrew Ng"
        }
    ],
    "start_index":3,
    "result_count":2
}
```


Figure 3: Data provided from ArtnetMiner services

With the result is given under html format, we implemented a program that can capture an author profile including id, name, phone, fax, email, homepage, position, affiliation and address. That information is partially considered for metrics that we implemented in our project. The data, however, is incomplete with some missing information for particular criteria of author profile and it is one of the limitations of this project.

In this project, the database contains 134,307 authors in total and 570,744 links represent for pair-collaborations within the period years 2000 - 2013. For a network of co-authorship, the graph G represents for it defined as following way. Let $G = (V, E)$ be an undirected graph with nodes $v_i \in V$ and edges $(v_i, v_j) \in E$, $1 \leq i, j \leq |V|$ where node represents for author and edge represent for collaboration. In co-authorship networks, self-links and directed edges are quite meaningless (unless there is metric requires edge direction; however, an undirected edge can always be converted into a pair of directed edge).

To test the accuracy of the framework, we split the data into two non-overlapping periods of time. We use the first set of data from 2000 to 2010 for training and the remaining

set of links after 2010 for testing so that the testing data size will be around 10% of the training data size. It should be noted that some authors only start publishing after 2010 and they doesn't appear in any collaborations before that. We will not consider those authors into the graph since that is beyond our scope. Hence we only keep nodes of researchers that presents in both time periods. After removing those authors, our graph now consists of 104,265 nodes of authors. These authors form 413,691 links in the period 2000 to 2009. After 2009, there are 35,558 new collaborations among them as described in table 2.

	2000-2009	2010-2013	Total (2000-2013)
No. of nodes	104,265	104,265	104,265
No. of links	413,691	35,558	449,249

Table 2: Co-authorship graph components

2.2 Features from previous work

In our work, we consider a variety of features mentioned in previous works that are presenting the state-of-art in link prediction. We list them in the below.

- Local similarity:
  - Common neighbours

"Common neighbours" directly counts the number of neighbours that the two nodes have in common. E.g., for two researchers in a collaboration network, this is the number of other researchers, with which both have had some collaboration. Clearly, if two researchers tend to collaborate with the same group of other researchers then they are likely to collaborate with each other as well.

  - Jaccard's coefficient

"*Jaccard's coefficient"* is a normalized measure of common neighbours. It computes the ratio of common neighbours out of all neighbours. This is sometimes a better measure than the *common neighbours*, especially when one node has a substantially larger neighbourhood than the other.

  - Adamic/Adar [3]

"*Adamic/Adar"* measures similarity between two nodes by assigning more weight to less shared common neighbours. It explained that two nodes which have a common neighbour that no other node has are often more similar than two nodes whose common neighbours are also common for many other nodes.

- Preferential attachment

"*Preferential attachment"* states that new links will be more likely to form among higher-degree nodes than lower-degree nodes. In a co-authorship network, this means a new collaboration is more likely to occur between authors who have a broader collaborative network.

- Keyword matching (extract from titles of the papers)

"Key words matching" evaluates the similarity of two authors based on their published paper. It could make sense since the papers indicate authors' expert area which is the basis for much collaboration.

- Global similarity:
  - Shortest path

The "*shortest path"* between two nodes is defined as the minimum number of edges connecting them. If there is no such connecting path then the value of this feature is generally assumed to be infinite. For many types of networks, the shorter the shortest path between two nodes is, the more likely the nodes are to become linked.

- PageRank

"PageRank" defines the probability that a node will be reached through a random walk on the graph. $P(a) = (1-d) + d\sum_{v\epsilon\Gamma(a)}\frac{P(v)}{|\Gamma(v)|}$, where $\Gamma(a)$ is the set of neighbourhoods of a; d is damping factor indicates the probability to visit the neighbour nodes rather than restart to the original node *a.* Since we need the ranking over pairs, we sum up page ranks of the two nodes. It can be explained as the chance in which two authors would likely be randomly picking each other as their partner. A modification of this feature is considering only the maximum or minimum value between two nodes. In this case, it can be viewed as the probability that the more prolific author (for maximum value case) or the less prolific author (for minimum value case) would likely to work with the other.

- Other baseline features

To further research and fulfil the baseline requirements for our experiment, we implemented following features on our own:

  - Katz[18]
  - Second Shortest Path[17]
  - SimRank[16]

Since those features have high complexities, it makes implementation really challenging and time consuming .With our implementation, we can only use those feature with a small-scale network.

2.2.3 Our proposed features

From previous papers, we observed that the features vectors are extracted mainly based on topology network. As describe, we are interesting in using the author profile and predicting the future link as well. Our contributions in this area will be introduce following metrics for link prediction.

- Productivity

Observing the data from co-authorship publication, we propose a feature that captures the productivity of the author over time. Within a period of time, we want to observe how fruitful an author can be. Some authors can have a tendency to expand collaborations with other, while some authors still keep a steady state of productivity. Also some particular authors might be used to co-author with numerous authors within the earlier period of the training state, but after that they "retire", in other words, they stop collaborating with the others. Here, we introduce the estimation of our metric to measure productivity of author. For an author node A, the productivity metric is described as:

$$P_A = \sum_{k=0}^{n} \frac{[n(k+1)-n(k)]^{m(K+1)-m(k)}}{T(k+1)-T(k)}$$

where *n(k)* denotes the number of papers at year *k* ; *m(k)* denotes the number of neighbours of A at year *k* and *T(k)* simply denotes the year *k* . This metric follows an intuition that within a period of time, an author who publishes more papers with new authors will have higher score than an author who publishes more paper but with the same old colleagues (authors that has been collaborated before) and clearly score higher than author who still haven't published any paper yet. Over time, there are many periods of time that events of co-

authorship could happen and the productivity of an author is accumulated over all the periods $T(k+1) - T(k)$. However, the importance of each period should also be considered, the latest co-authorship events should be weighted higher than the earlier events. Therefore, a weighted formula can be:

$$P_A = \sum_{k=0}^{n} \frac{\alpha^k . [n(k+1) - n(k)]^{m(K+1)-m(k)}}{T(k+1) - T(k)}$$

, where α is a constant (α > 1) that will increase every time an event of co-authorship is triggered. To compute the feature parameter for a pair of node (A, B), we simply sum up the productivity values $P_{A,B} = P_A + P_B$.

- Common Third Neighbour

Prefer to Common Neighbour metric above, this metric is our modification in which suggest the exploring the "*area of influence*" of authors 3-paths away from the original vertex. With the same idea of the Common Neighbour but on a larger scale of prediction, this metric has been proved to work more effectively under our network scope. In fact, we did a small experiment to compare among Common Neighbour, Common Second Neighbour and Common Third Neighbour (prefer to Appendix A). The result shows that the more we increase the author's neighbourhood path, the more possible collaborations (recall value) we obtain.

- Address

This is the university address in which the author is working. The address reveals the author geography location. It would be the best for author in same institute or university to collaborate as usual. Somehow, the geography also plays part of a role in the decision whether the author want to pair with the other or not. And so we use the Jaccard Similarity here to capture the likelihood of the authors' addresses.

- Affiliation

"*Affiliation*" would depict an author position, for example: Department of Computer Science and Engineering which is what we retrieve from our data set. This is a simple metric in which decide if two authors would likely to cope with each other based on their affiliation otherwise, we won't recommend collaboration for cross-department.

- Keyword

"*Keyword*" here is not the actual keyword that the papers provide since our dataset don't have enough information for this. We decided to extract the keyword from all the title papers that belong to the owner. With a bag of keyword tokens, we also use Jaccard Similarity to measure the likelihood. This metric reflects the field of research and would describe the collaboration among authors with same interests. In recent research papers, keyword is considered as a factor of link prediction with different modifications of the likelihood measurement.

- Conference

The data from the author profile describe which conference that particular papers of his or her was published to. Conferences are places where authors know each other especially for international researchers, where scientists look for new thesis and ideas. Therefore, it would be an interesting environment for authors getting to know each other. The idea also describes that the person who participate many conferences tend to be more willing to collaborate.

With all above metrics about author profile, we expect to obtain a more concise link prediction in the future. However, as we mentioned, the dataset is incomplete with many missing data from the author profile. This will lead to the incomplete learning and wrong predictions.

III. EXPERIMENT

We first identified two state-of-art papers ([6], [2]) as the baselines. These two papers tackle the same problems as us, but use different dataset. [6] uses DPLP data source (http://www.informatik.uni-trier.de/~ley/db/) while [2] uses Institute of Electronics Information and Communication Engineers (IEICE) dataset (http://www.ieice.org/eng/index.html). To make it more comparable, we will use our own data and test it using their specific feature set as well as predictors. It should be noted that this is just a relative comparison as many factors should be taken into account such as parameters of the predictors or the co-efficient of some of the metrics the two authors utilized.

3.1 Experimental set up

For *n* nodes in the network, there will be $N = \frac{n \times (n-1)}{2}$ vectors. To predict a link, we partition the range of publication years into two non-overlapping sub-ranges as we discussed in the method earlier.

Now, we define two labels true/false for the vectors according to the existent/nonexistent link within the period of training, 2000 -2010.

For training set denoted by set *S*, we pick out all the true labels (S_T) and randomly choose a number of false labels (S_F); $S = S_T \cup S_F$. The reason that we only choose a limited number of false labels is because we observe that the number of false labels is too big in compare with the number of true labels, which might lead to bias prediction in training.

Testing is done on the remaining set $R = N / S_T$ which is the set of nonexistent links in period 2000-2009 but appear in period 2010 – 2013. Each such pair either represents a positive example or a negative example, depending on whether those author pairs published at least one paper in the testing years or not. The accuracy of a predictor is estimated by comparing the result labels with the set *R*.

Extracting values for some of the features, especially, features depending on network structure, on a large scale network is time consuming. For instance, it takes around 2 seconds for getting a shortest path value for only one pair of authors. This could probably the most

challenging issue of this project since it makes debugging and implementing the program more difficult. Thank to WING powerful server, we can manage to get the data results. However, due to resource limitation as well as time restriction, there are features that we can't manage to obtain the result which are SimRank mentioned in the work of P.Milen et al. [2] and Second Shortest Path mentioned in the work of M. Al Hasan et al. [6]. In previous work of P.Milen et al.[2], the authors experiment on a small co-authorship network which is around 1,700 authors only and hence the SimRank value can be easily computed. For Second Shortest Distance, we did implement and debug it; however, due to time restriction, we are forced to add it into future work.

- Classification algorithms

There exist plenty of classification algorithms for supervised learning. Although their performances are comparable, some usually work better than others for a specific dataset or domain. In this research, we experimented with classifiers including Decision Tree, Logistic, Naives Bayesian and SVM from the baselines. We use WEKA [19], a well-known machine learning library, to conduct experiments.

3.2 Experiment Result

First of all, we compare the baseline with our choosing feature set. The first one is from M. Al Hasan et al. paper [6] (baseline 1). The features using in this work include:

- Shortest Path
- Sum of Paper
- Common Neighbor
- Second Shortest Path

According to his paper selection of classifiers, we collected the following result (Table 3).

	Precision	**Recall**	**F-value**
Decision Tree	0.98	0.37	0.53
Bagging	0.98	0.39	0.56
SVM(Linear Kernel)	0.85	0.3	0.45
Naives Bayesian	0.98	0.54	0.69
Multilayer Perceptron	0.87	0.36	0.5

Table 3: Baseline 1 performance implement based on our dataset

For the second paper of P.Milen et al. [2](baseline 2), the features set include:

- Shortest Distance
- Common Neighbors
- Jaccard's coefficient
- Adamic Adar
- Preferential attachment
- Katz
- PageRank (min)
- PageRank (max)
- SimRank

The authors use following classifiers (Table 4):

	Precision	**Recall**	**F-value**
SVM	0.99	0.29	0.45
Decision tree	0.99	0.39	0.56

Table 4: Baseline 2 performance implement based on our dataset

As we observe, the best performance from the previous baseline is Naïve Bayesian (69.8%) and the best performance from the latter baseline is Decision Tree (56.3%). However, different classifiers are not expected to perform equivalently. Therefore, we consider only SVM and Decision Tree which are the common classifiers that the two authors used with almost equivalent settings.

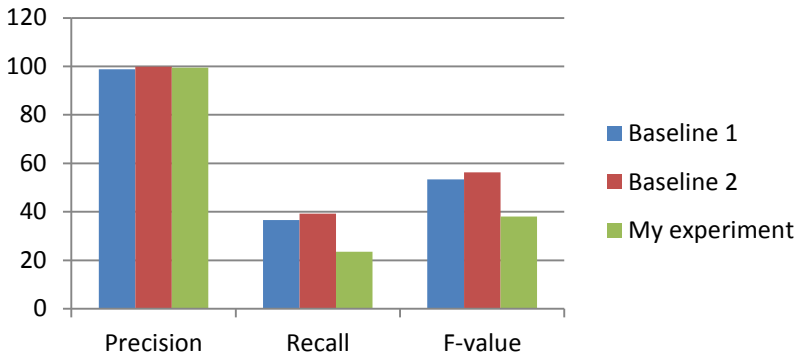


Figure 4: Comparison among two baselines and original feature set using Decision Tree learning

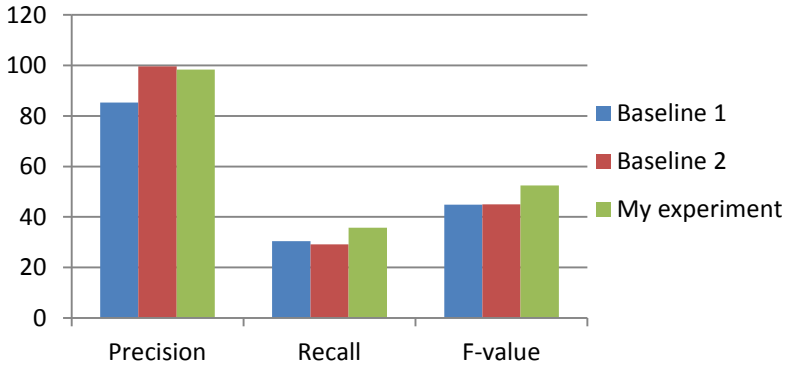


Figure 5: Comparison among two baselines and original feature set using SVM learning

Firstly, we noted that baseline 1 and baseline 2 taking different feature sets but end up with almost same performance which can possibly infer that some of the features cover the same information. In other words, some of them are not really influence the overall performance if we remove it. Besides, there is a fact that with the information from training data, we can predict almost all the links in the testing domain (since over 90% precision obtained from all the performances). From the previous corresponding works, precision is varying from 70% to 85%; however, there is no much thing to say since they are using different datasets. Secondly, considering our own set of features has resulted in worse performance in decision tree but better in SVM, we are eager to find out the reason so that in the end we could improve the F-value of decision tree and even SVM.

Ranked Attribute	Attribute Name	Info Gain
1	Conference	0.96
2	Shortest Path	0.95
3	Adamic Adar	0.93
4	Keyword	0.80
5	Common Third Neighbor	0.43
6	Preferential Attachment	0.33
7	Sum of Paper	0.26
8	PageRank	0.25
9	Productivity	0.17
10	Affiliation	0.03
11	Address	0.02

Table 5: Information Gain Ranking of features

According to Table 5, among the attributes we recommended, the feature "*Conference*" is proving quite informative while the others such as "*Affiliation*", "*Address*" are determined as the least useful of all with 2% to 3%. It is expected that the information gain from those features would not be so high since we actually miss a bunch of author profile data as we have mentioned in the previous. We could easy see from the data distribution of "Address" in the Appendix that almost 99% of the value is 0 which indicates that there is no information about the address to compute the similarity score. However, what surprise us is the "*Conference*" attribute which is ranking on top of all can be a good "candidate".

Digging into the problem, we observe the distribution of positive and negative samples (prefer to Appendix figures) and further experiment on each individual attributes to observe their behaviors. With each of the metrics, we will also use Decision Tree and SVM to classify.

	Precision	Recall	F1
Adamic Adar	0.99	0.29	0.44
Common 3rd Neighbour	0.85	0.71	0.77
Conference	0.97	0.32	0.48
Address	0.92	0.06	0.11
Key words	0.89	0.45	0.6
Productivity	0.77	0.73	0.75
Preferential Attachment	0.78	0.71	0.74
PageRank	0.95	0.29	0.44
Affiliation	0.67	0.71	0.69
Shortest Path	0.99	0.29	0.44
Sum of paper	0.78	0.75	0.76

Table 6: Performance of individual feature classified by Decision Tree

	Precision	Recall	F1
Adamic Adar	0.99	0.18	0.3
Common 3rd Neighbour	0.9	0.58	0.7
Conference	0.97	0.29	0.45
Address	0.92	0.06	0.11
Key words	0.79	0.3	0.44
Productivity	0.83	0.68	0.75
Preferential Attachment	0.98	0.18	0.31
PageRank	0.85	0.5	0.63
Affiliation	0.67	0.71	0.69
Shortest Path	0.88	0.49	0.63
Sum of paper	0.85	0.6	0.7

Table 7: Performance of individual feature classified by SVM

In general, all the features tend to make quite high precision but return rather low recall. Usually, there is a tradeoff between precision and recall, so in order to evaluate each

feature performance in this experiment, we will base on their F-value result. Attributes such as "*Common Third Neighbor"*, *"Affiliation"*, *"Productivity"*, *"Preferential Attachment"* and *"Sum of Paper"* outperform the others from both classifiers since they have not only high precision but also good recall. We also pay attention to features that we have recommended. Apart from "*Common Third Neighbor", "Affiliation"* and *"Productivity",* the "Keyword", "Address" and "Conference" features seem failed to meet our expectation at least under this project domain, so further modification on those features can be added up to future work.

With the results obtained from above, we select 4 most well performed features and used it as the final feature set including: Common Third Neighbor, Preferential Attachment, Productivity and Sum of Paper.

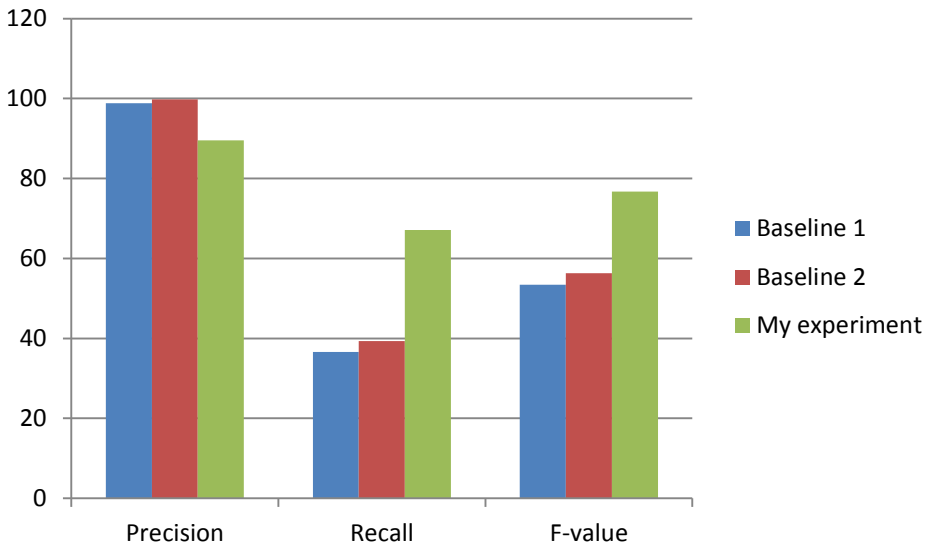


Figure 6: Comparison among two baselines and final feature set using Decision Tree learning

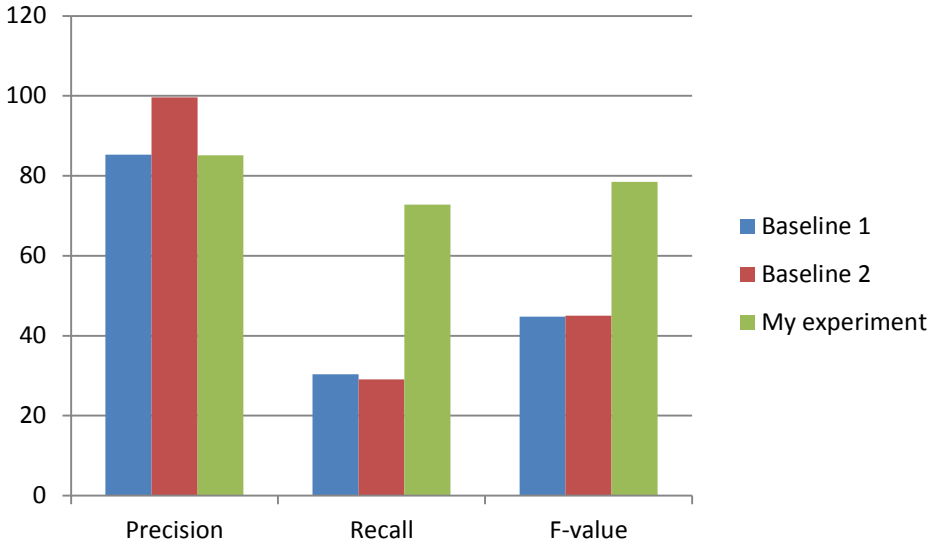


Figure 7: Comparison among two baselines and final feature set using SVM learning

The final result not only outperforms the two baselines with Decision Tree learning (over 20% F-value) but also with SVM learning (over 30% F-value). Although the precision is actually decreased, its recall is indeed higher than the previous set up. This can be considered as the best performance that we obtained. In fact, we also notice that using one of the above attributes alone is also getting us a huge boost up. For example, with the "*Common 3rd Neighbor*" metric, the F-value is already 70% for SVM in compare with 77% of the final result.

IV. CONCLUSION

4.1 Summary

In this report, we have introduced supervised learning approach for link prediction in co-authorship network. With the focus on features set, we have recommended the "*Productivity"* metric which combines not only the network structure but also the publish data of the researcher and it has proved to be useful for link prediction in co-authorship network. We also suggest some attributes taken from author profile data in addition to the common network-structure metrics. Among those, the "*Affiliation*" is comparable with the other old metrics. So there are two out of five related author profile are actually works.

Experiment is set up almost similar to the baselines but with our own implementation. We tried to follow their work specification, however, there are still some slightly modifications in the metric algorithms so that it can suit to our network. Besides, there is vague information in the baseline such as specification on number of training data or constant values in metric algorithm, iterations on PageRank, etc. so we decide to adjust with our specifications. Nevertheless, our experiment can still be a comparable representation for the baseline works.

4.2Limitation

As mentioned above, the limitation of this project is the incomplete database. Originally, this project targeted to utilize personal information of researchers as to distinguish from previous works since most of the related works depend on building the network structure for link prediction. Researchers do not usually provide their personal data on the database library, but rather they would have preferred putting links to their homepages. That is why we prefer mining data from multiple sources which might have highly structured database.

Another significant problem for supervised learning highlighted by Lu, L. et al [9] is that the approach can be prohibitively time consuming for a large network (network contains more than 10,000 nodes). In fact, extracting feature vectors are one of the step that taking much time. Besides, implementation features algorithm to run on a large network requires higher optimizations and more memory resources, especially, for features related to global network structure such as Katz and SimRank.

4.3Further work

Aroused from the limitation of this project, an optimization of the algorithm for Katz, SimRank attributes should be taken into account. Although we managed to make it work on a small-sized network, it still can't run on large network yet. Besides, there can be many modifications on the author profile attributes that can be considered. We could learn whether an author is a PhD student or a Professor and based on that predict the tendency of working. A recent paper also has the idea of classification a link in co-authorship network into teacher-student or student-student roles. To further the prediction of collaboration, we might even look on age of the authors, graduation year, etc. as long as we can retrieved those information. After all, there is still a lot of value information that can be mining from author profiles to serve the prediction judgments.

REFERENCES

[1] Liu, X., Bollen, J., Nelson, M. L., & Van de Sompel, H. (2005). Co-authorship networks in the digital library research community. Information processing & management, 41(6), 1462-1480.

[2] Pavlov, M., & Ichise, R. (2007). Finding Experts by Link Prediction in Co-authorship Networks. *FEWS*, *290*, 42-55.

[3] Adamic, L. A., & Adar, E. (2003). Friends and neighbors on the web. *Social networks*, *25*(3), 211-230.

[4] Clauset, A., Moore, C., & Newman, M. E. (2008). Hierarchical structure and the prediction of missing links in networks. Nature, 453(7191), 98-101.

[5] Wang, C., Satuluri, V., & Parthasarathy, S. (2007). Local probabilistic models for link prediction. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*,322-331.

[6] Al Hasan, M., Chaoji, V., Salem, S., & Zaki, M. (2006). Link prediction using supervised learning. In *SDM'06: Workshop on Link Analysis, Counter-terrorism and Security*.

[7] Getoor, L., & Diehl, C. P. (2005). Link mining: a survey. *ACM SIGKDD Explorations Newsletter*, *7*(2), 3-12.

[8] Murata, T., & Moriyasu, S. (2008). Link prediction based on structural properties of online social networks. *New Generation Computing*, *26*(3), 245-257.

[9] Lü, L., & Zhou, T. (2011). Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, *390*(6), 1150-1170.

[10] Guimerà, R., & Sales-Pardo, M. (2009). Missing and spurious interactions and the reconstruction of complex networks. *Proceedings of the National Academy of Sciences*, *106*(52), 22073-22078.

[11] Lichtenwalter, R. N., Lussier, J. T., & Chawla, N. V. (2010). New perspectives and methods in link prediction. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, 243-252.

[12] Soares, P. R., & Prudêncio, R. B. (2013). Proximity measures for link prediction based on temporal events. *Expert Systems with Applications*, *40*(16), 6652-6660.

[13] Cohen, S., & Ebel, L. (2013). Recommending collaborators using keywords. In *Proceedings of the 22nd international conference on World Wide Web companion* 959-962.

[14] Liben-Nowell, D., & Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the American society for information science and technology*, *58*(7), 1019-1031.

[15] Brandão, M. A., Moro, M. M., Lopes, G. R., & Oliveira, J. P. (2013). Using link semantics to recommend collaborations in academic social networks. In Proceedings of the 22nd international conference on World Wide Web companion, 833-840.

[16] Jeh, G., & Widom, J. (2002, July). SimRank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 538-543). ACM.

[17] Eppstein, D. (1998). Finding the k shortest paths. *SIAM Journal on computing*, *28*(2), 652-673.

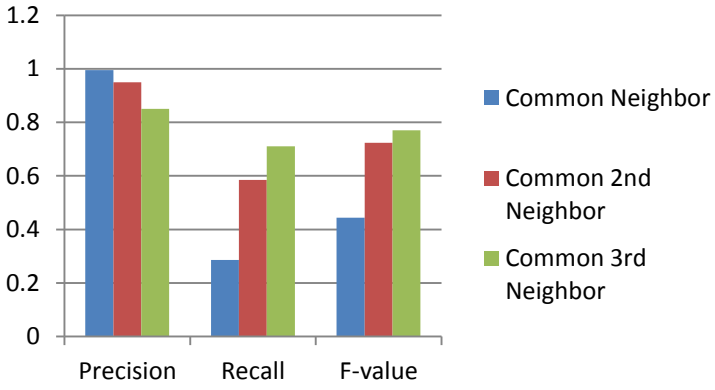
[18] Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika*, *18*(1), 39-43.

[19] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, *11*(1), 10-18.

APPENDIX A

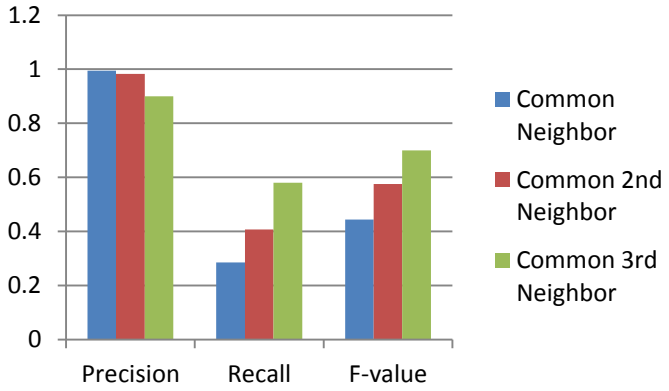
Experiment on Common Neighbor, Common Second Neighbor and Common Third Neighbor features.

	Precision	Recall	F-value
Common Neighbour	99.5	0.286	0.444
Common 2nd Neighbour	95	0.585	0.724
Common 3rd Neighbour	85	71	0.77



Using Decision Tree to classify

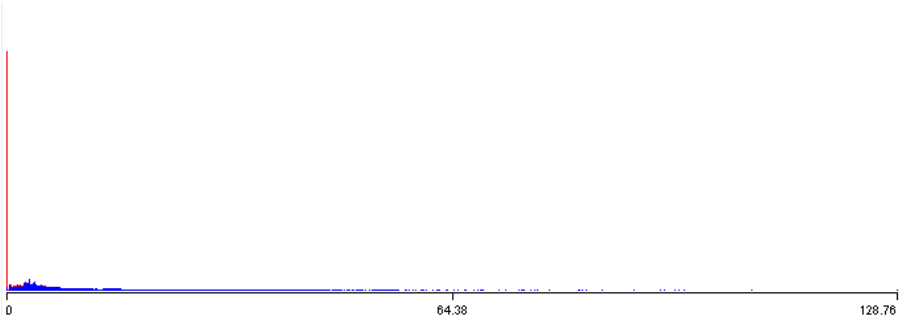
	Precision	Recall	F-value
Common Neighbour	0.995	0.286	0.444
Common 2nd Neighbour	0.983	0.407	0.575
Common 3rd Neighbour	0.9	0.58	0.7



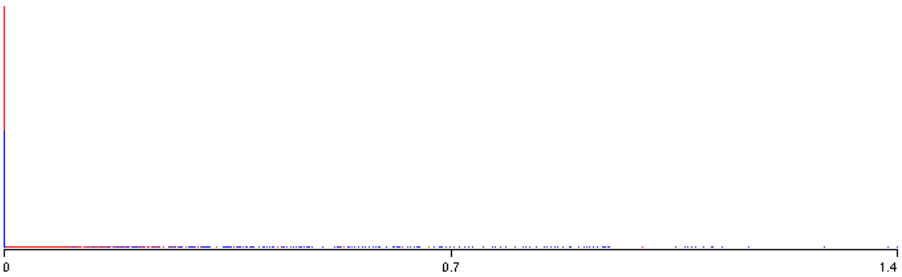
Using SVM to classify

APPENDIX B

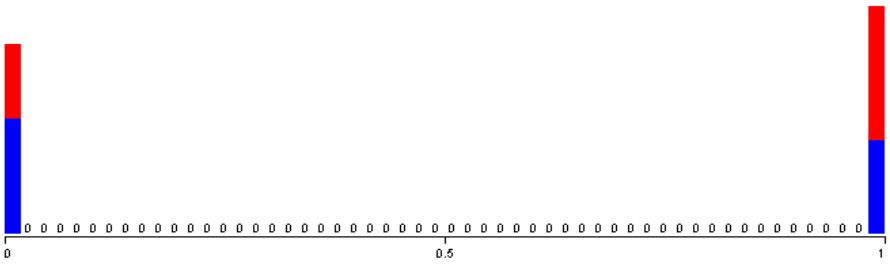
The data distribute of the features in experiment



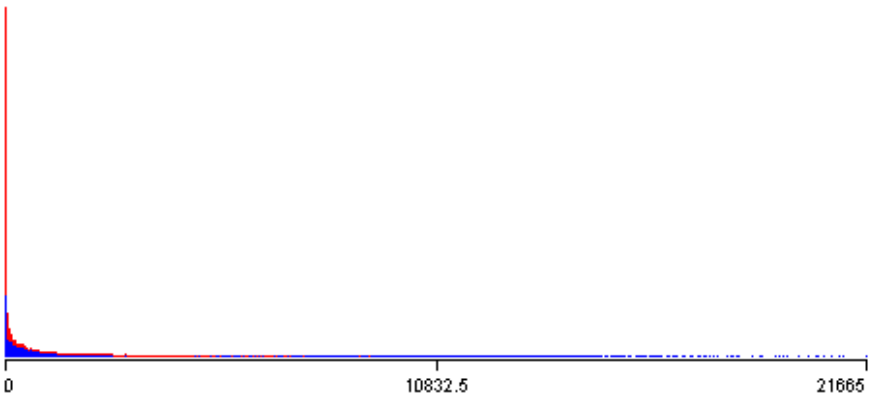
(1) Adamic Adar



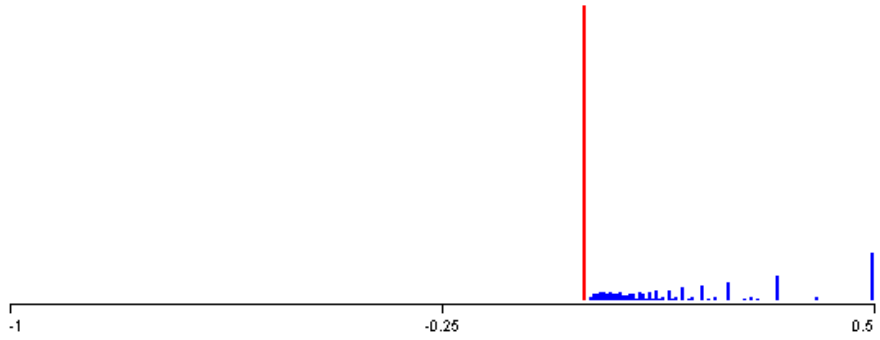
(2) Address



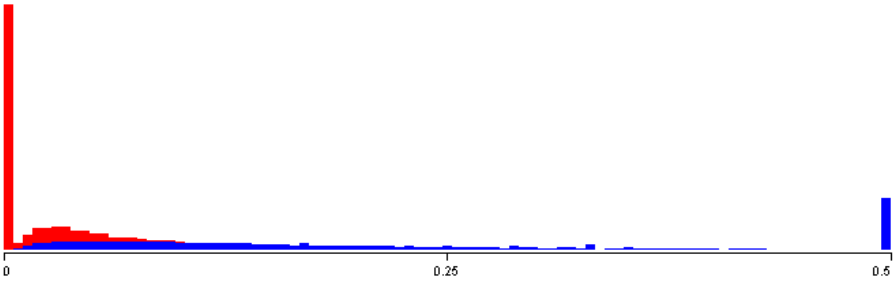
(3) Affiliation



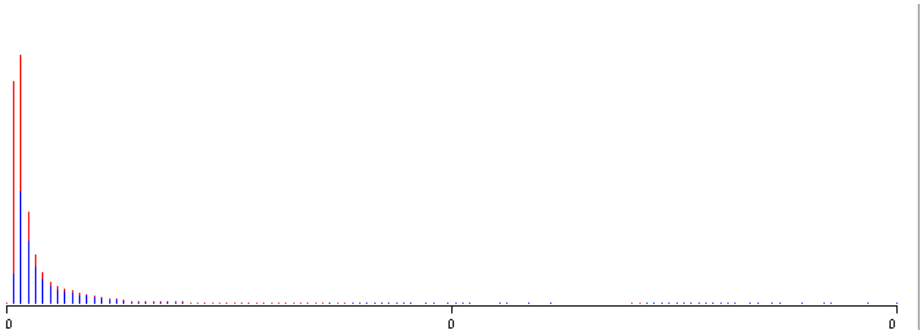
(4) Common Third Neighbor



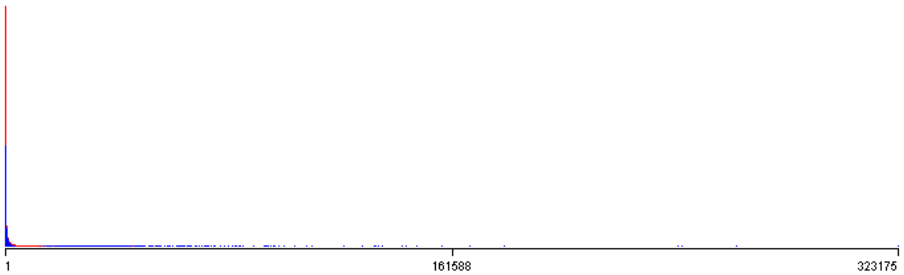
(5) Conference



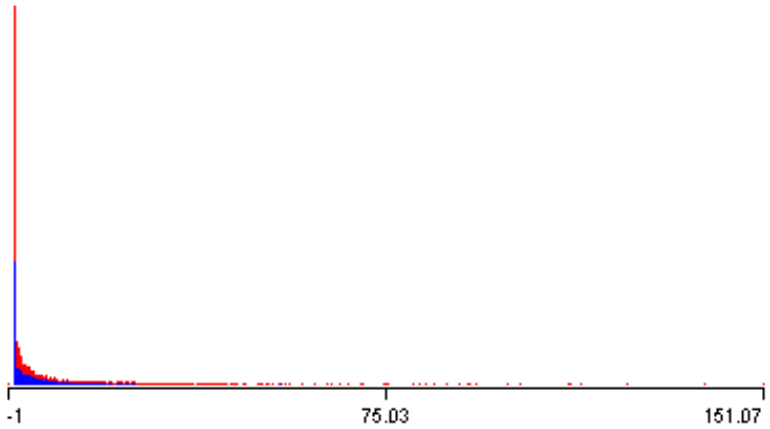
(6) Keyword



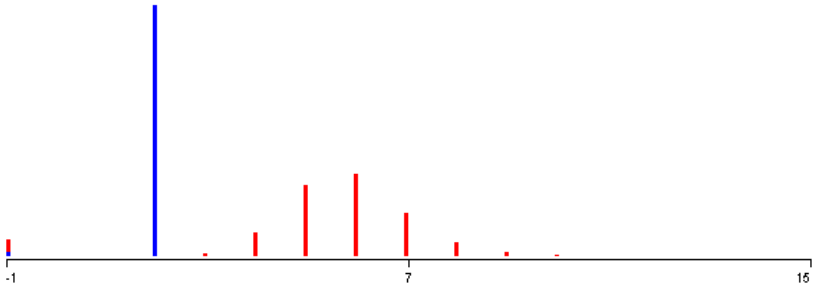
(7) PageRank



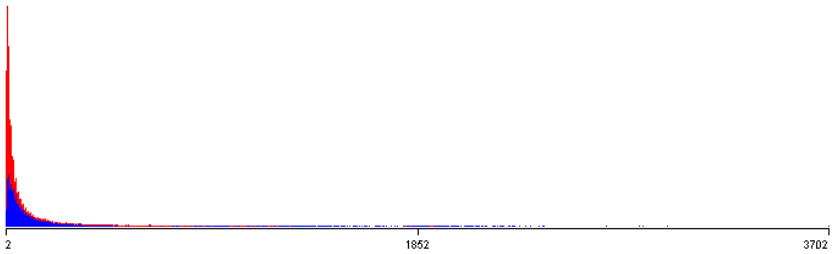
(8) Preferential Attachment



(9) Productivity



(10) Shortest Path



(11) Sum of Paper