Universiteit Leiden
Opleiding Informatica

An Evaluation Method
for Nodes
in Multiple Dynamic Networks

Name: Wei Liu
Date: 20/05/2015
1st supervisor: Dr. W.A. Kosters
2nd supervisor: Dr. F.W. Takes

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

An Evaluation Method for Nodes in Multiple Dynamic Networks

Wei Liu
LIACS
Leiden University
evenfall@live.com

Abstract

With the rapid development of internet and information technologies, ranking methods only based on simple citation (link) networks are already insufficient for the growing requirement of information extraction and evaluation. We propose a method for evaluating the influence and importance of nodes that are contained in multiple dynamic networks. This method combines a modified PageRank algorithm and modern data mining techniques such as a community detection algorithm and artificial neural networks (Self-Organizing Maps, SOMs). A citation network from the e-print Arxiv HEP-TH (high energy physics theory) is processed to serve as main dataset in the experiment, as well as data from social networks such as Facebook and Twitter. Some clustering results and visualizations, which indicate the influential trends (through time) of both individual nodes in a network and the network itself, are produced using different configurations.

1 Introduction

The citation (link) graph of the web is an important resource that is commonly used for ranking the search result from web search engines since the PageRank algorithm came out [11]. However, with the rapid development of internet and information technologies, the citation graphs are getting more complicated and contain more and more information everyday. Ranking methods only based on simple citation already cannot meet the growing requirement for information extraction and evaluation.

On the other hand, the data mining techniques have a significant development as well. The analysis focused on clustering information networks into detailed and precise categorized communities is a very hot topic and already gained a lot of achievements [15]. Therefore, the community attribute of citation networks should also be taken into account when evaluating them. Not only the influence of nodes within communities, but also the influence cross multiple communities.

In this paper, a dynamic evaluation method, which can be applied to datasets with multiple dimensions, is developed in order to generate ranking and trend information in citation networks with community attributes. This method combines the simplicity and efficiency of the PageRank algorithm and modern data mining methods such as community detection methods and neural network techniques. This method is a dynamic method because the evaluation results will evolve when there are new input data. In order to apply our method to datasets with multi-dimensional attributes, we also created a data sturcture called Multiple Dynamic network structure.

Therefore, the research questions of this thesis are:

- Build the MDN structure from various data sources.
- Develop a general method to evaluate the target network in the MDN structure.

Evaluating the influences of nodes in the target network and the trend information of the target network itself are the main subject of this thesis.

Section 2 introduces the main research concept and workflow. Section 3 describes the datasets used in this thesis. Section 4 lists some other papers related to our research. Section 5 demonstrates the methods and algorithms used in our research. Section 6 illustrates the whole experiment process and analyses the results. Section 7 is the conclusion and future work.

This research and is part of a master project within LIACS, the Computer Science Department of Leiden University. It is supervised by Dr. Walter Kosters and Dr. Frank Takes.

2 The Concept and Workflow

Before getting into the details about the research methods and experiments, the main concept, which is the Multiple Dynamic Network structure, and the general workflow are introduced in this section.

2.1 Multiple Dynamic Network Structure

We first introduce the Multiple Dynamic Network (MDN) structure, see Figure 1 for example. The first thing to be noticed in this example figure is that there are multiple network layers (networks N_1, N_2 and N_3) in the structure. The number of the network layers is variable. Each layer represents a network, which is categorized by the same categorizing rules from the same data source. The categorizing rules means the rule to divide communities for the data source. For example, if the data source is the user relationship network from Facebook, the categorizing rule could be “What are the users

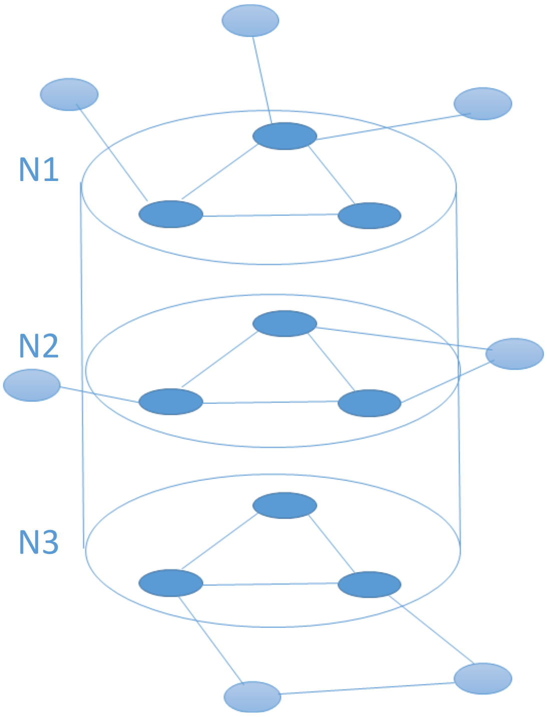


Figure 1: The general example for the MDN structure: N_1, N_2, N_3 are three different networks from the same data source, but they share the same sub-network (the triangle shaped network within the circle).

interested in?". Then the network N_1 could be Facebook users who are interested in food, network N_2 could be people who are interested in sport, and so on. If the data source is the citation network of scientific papers, the categorizing rule could be "When are the papers published?". In other words, the categorizing rule is an attribute from the data source which can be used to categorize individual instances from the data source.

If we want to build the MDN structure, the categorizing rules should be chosen carefully in order to produce the essential part of the MDN structure: the target network. The target network is a sub-network contained by all categorized networks, which is the network with all the darker nodes from Figure 1. For this general example of the MDN structure, networks N_1, N_2 and N_3 represent three communities which are categorized by a certain categorizing rule. They share the same target network, which contains three nodes. In other words, the categorized networks of the MDN structure are different from each other, but the sub-network within the circles, which are the target network, are exactly the same.

2.2 The General Workflow of the Method

The general workflow of our evaluation method is summarized in Figure 2. The detailed information about the research methods mentioned in the workflow are explained in Section 5. How to develop and use this workflow is explained in Section 6. The start point of the workflow is pre-processing the raw dataset, which is formatting the dataset into the MDN structure. Applying the categorizing rule is necessary for this part unless the dataset is already MDN formatted. Because the variety of different datasets, abstracting a general method to format datasets into the MDN structure is a very complicated task, which is not the major goal of this thesis. Therefore, during our experiments, we applied different pre-processing methods to individual datasets according to their attributes.

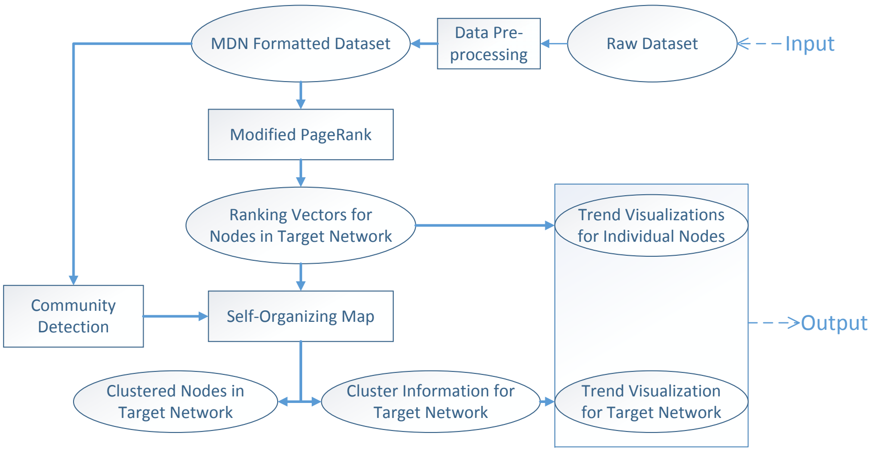

MDN Formatted Dataset
Data Pre-processing
Raw Dataset
Input
Modified PageRank
Ranking Vectors for Nodes in Target Network
Trend Visualizations for Individual Nodes
Community Detection
Self-Organizing Map
Output
Clustered Nodes in Target Network
Cluster Information for Target Network
Trend Visualization for Target Network

Figure 2: The general workflow of the evaluation method

After the dataset pre-processing, the MDN formatted dataset will be the input for the modified PageRank algorithm, which will be introduced in Section 5.2. The ranking vectors for the nodes in target network, which are vectors containing the ranking values for the target network nodes in all categorized networks $N_1, N_2, \ldots, N_q$ (where q is the number of categorized networks), is generated as output of this part of workflow. The output of this part can be used for trend visualization for individual nodes, also functioning as the input for the Self-Organizing Map (Section 5.4). Together with the assistance from community detection method (Section 5.3), the Self-Organizing Map can produce the cluster information for the target network and nodes in the target network. Finally, the cluster information for the target network will be visualized to evaluate the trend information of target network.

3 Datasets

In this section we discuss the type and the source of the datasets used in our experiments.

3.1 The Type of Datasets

The datasets used in this thesis are all in graph [1] type, which is a representation of a set of objects where some pairs of objects are connected by links. The interconnected objects are called nodes, and the links that connect some pairs of nodes are called edges. The edges may be directed or undirected. Both nodes and edges can be assigned to some value, such as a symbolic label (ID, name, etc.) and/or a numeric attribute (weight, cost, length, etc.).

3.2 The Source of Datasets

The source of datasets is the Stanford Large Network Dataset Collection (SLNDC) [2]. The first dataset used in our experiments consists of "friends lists" from Facebook [3]. The dataset includes node features (profiles), friends lists, ego networks and the edge network of all egonets combined. The second dataset consists of "circles" from Twitter [4], and shares the same basic structure with the first one. The only difference is that the edges of the former are undirected, the later are directed. These two datasets have been anonymized by replacing real IDs for each user with a new value. Also, while feature vectors have been provided, the interpretation of those features has been obscured. The third dataset used in our experiments is the Zachary karate club network [5], which was collected from the members of a university karate club by Wayne Zachary in 1977. Each node represents a member of the club, and each edge represents a tie between two members of the club. The fourth dataset used in the experiment is from SLNDC as well, and is a citation network from the e-print Arxiv HEP-TH (high energy physics theory), the dataset covers papers in the period from 1993.1 to 2003.4 (124 months). Each node represents a paper in a journal, and each edge represents a citation from source to target. Table 1 shows statistics for the datasets mentioned above.

Dataset Statistics	Nodes	Edges	Diameter	clustering coefficient
FACEBOOK [3]	4,039	88,234	8	0.6055
TWITTER [4]	81,306	1,768,149	7	0.5653
KARATE [5]	34	78	5	0.256
HEP-TH [6]	27,770	352,807	13	0.3120

Table 1: Atrributes and statistics for datasets.

4 Related Work

Our work in this paper focused on an evaluation method, which use the PageRank algorithm for evaluating the nodes' influences on citation networks. We found the papers below are related to our work.

4.1 PageRank on Citation Network

Ding et al. [12] studied how varied damping factors in the PageRank algorithm can provide additional insight into the ranking of authors in an author co-citation network. They calculate the ranks of these 108 authors based on PageRank with damping factor ranging from 0.05 to 0.95. They found out that citation rank is highly correlated with PageRank's with different damping factors. They also found out that both popularity rank and prestige rank were highly correlated with the weighted PageRank [13]. Their research pointed out that the Pagerank algorithm is suitable for citation networks of scientific papers, but does not incorporate time feature.

4.2 Evaluation Methods of the Nodes' Influences

Luiten et al. [15] sampled a portion of the Twitter social graph, from which they have distilled topics and topical activity, and constructed a set of diverse features. They found out that only looking at simple popularity features such as the number of followers is not enough to capture the concept of topical influence.

Takes and Kosters [16] proposed an exact algorithm that uses various lower and upper bounds as well as effective node selection and pruning strategies in order to evaluate only the critical nodes which ultimately determine the diameter of small world networks. They also introduces a set of classification techniques for determining the difficulty (for a human) of path traversal in an information network [17]. They focus on local and global structural graph properties and measures to determine the difficulty of finding a certain path.

Kazienko et al. [18] evaluated the mutual interaction in social network of internet users based on the node positions. Weng et al. [19] and Wu et al. [20] also did research on how to find the influential twitterers and how and what they twittered.

5 Research Methods

In this section, all research methods that are used in this paper are indroduced. Including ranking method, community detection methods, and machine learning methods.

5.1 PageRank

PageRank is an algorithm first developed by Brin and Page [11] to rank websites in their search engine results. PageRank is a way of measuring the importance of website pages, as well as other citation networks. It works by counting the number and quality of links to a node to determine an estimate of how important the node is. The underlying assumption is that more important nodes are likely to receive more links from other important nodes. Figure 3 shows an example of how PageRank works [1]. What we see here is a network containing 11 nodes. Each node has various numbers of links to other nodes. However, the PageRank value not only depends on the number of links. Page C has a higher PageRank than page E, even though there are fewer links to C; the one link to C comes from an important page and hence is of high value. If web surfers who start on a random page have an 85% likelihood of choosing a random link from the page they are currently visiting, and a 15% likelihood of jumping to a page chosen at random from the entire web, they will reach page E 8.1% of the time. (The 15% likelihood of jumping to an arbitrary page corresponds to a damping factor of 85%.) Without damping, all web surfers would eventually end up on pages A, B, or C, and all other pages would have PageRank zero. In the presence of damping, page A effectively links to all pages on the web, even though it has no outgoing links of its own.

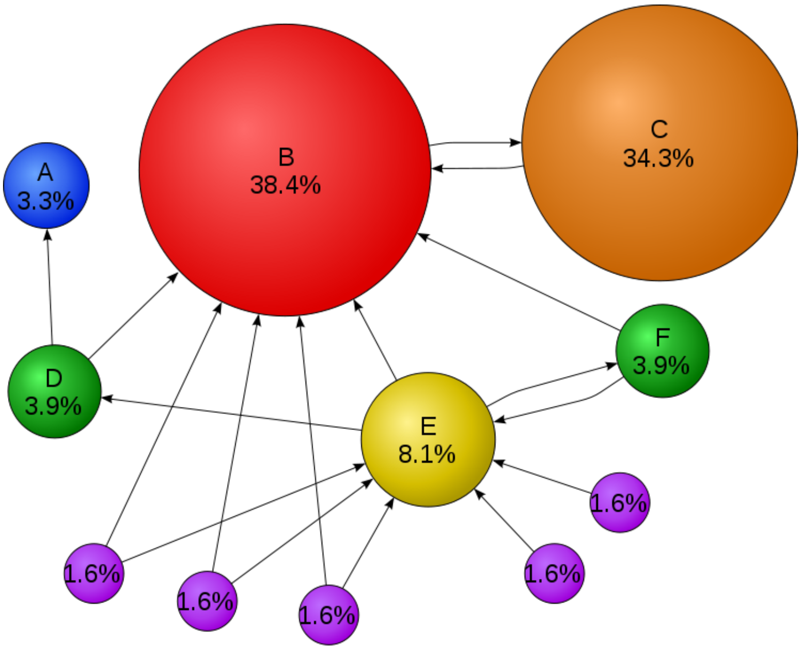


Figure 3: Example for PageRank. Image from Wikipedia [7].

[1] http://en.wikipedia.org/wiki/PageRank

5.2 The Normalization of PageRank

The PageRank algorithm works really well on citation networks for both aspects of function and efficiency. Therefore, the PageRank algorithm is selected as ranking method for the categorized networks in the MDN structure. To generate proper ranking vectors for the MDN formatted data set, the original PageRank algorithm needed to be modified to adapt to the dynamic condition of categorized network.

In order to describe the problem of regular PageRank and how to fix it, we denote the PageRank of a node n in a network N by $PR_N(n)$, and $|N|$ denotes to the total number of nodes in the network N. Therefore, the algorithm of PageRank [11] can be presented as:

$$PR_N(n) \leftarrow \frac{1-d}{|N|} + d\left(\frac{PR_N(n_1)}{L(n_1)} + \frac{PR_N(n_2)}{L(n_2)} + \cdots\right) \tag{1}$$

where d is the damping factor. Here the damping factor is set to be 0.85 [11], and $L(n)$ is the number of the outbound links for the node n or the number of the links between node n and its neighbor nodes, depending on whether the links are directed or undirected. The initial value of $PR_N(n)$ is $\frac{1}{|N|}$.

Due to the Equation 1, we can deduce that:

$$\sum_{n \in N} PR_N(n) = 1 \tag{2}$$

From Equation 2 we can see that it is not easy to compare the ranking value for the same node between a network and its sub-network because the ranking value for each node will generally decrease when the network gets larger. Even a certain node can have a higher actual importance value from a sub-network compared to the value from the whole network, the PageRank value can still remain unchanged or even get lower. This situation is really inconvenient for the information extraction and visualization. Therefore, in order to compare the Pagerank value of the same node n with different networks $N_1, N_2, \ldots, N_i$ with different sizes that all contain node n, here we introduce a method of normalization to eliminate this disturbing variation caused by the size difference of networks:

$$NPR_{N_2}^{N_1}(n) = PR_{N_1}(n) \times \frac{|N_1|}{|N_2|}, n \in \mathit{N_1}, \mathit{N_2}, \text{and} \mathit{N_2} \subseteq N_1 \tag{3}$$

Here $PR_{N_1}(n)$ is the PageRank value for node n from network N_1, and N_2 is a sub-network of N_1. Hence, $NPR_{N_2}^{N_1}(n)$ is $PR_{N_1}(n)$ normalized for network N_2, which can be used to compare with $PR_{N_2}(n)$ without any other interrupts due to the change of the size of networks.

Basically, in order to eliminate the variation of the PageRank value caused by size difference between two networks, this normalization method proportionally adjusts the PageRank value of a node, which exists in both network, based on the total number difference of nodes between these two networks.

5.3 Modularity Algorithm

Modularity [8] is a measure of the structure of networks. It was designed to measure the strength of the division of a network into modules (also called groups, clusters or communities). Networks with high modularity have dense connections between the nodes within modules, but sparse connections between nodes in different modules. Modularity is often used in optimization methods for detecting community structure in networks.

Modularity is defined as the fraction of the edges that fall within the given groups minus the expected such fraction if the edges were distributed at random. The value of the modularity lies in the range $[-1/2, 1)$. It is positive if the number of edges within groups exceeds the number expected on the basis of chance. For a given division of the network's vertices into some modules, modularity reflects the concentration of nodes within modules compared with a random distribution of links between all nodes regardless of modules.

The modularity algorithm [10], which we used as community detection algorithm in the workflow, is an algorithm dividing the network into communities by optimizing the modularity of the network. This algorithm find high modularity partitions of the network in short time, as well as allows for the access to different resolutions of community detection. During the experiments, we stick to the default resolution, which is 1.0, for the sake of consistence.

5.4 Self-Organizing Map

The *Self Organizing Map* (SOM), which is a type of artificial neural network, is an approach for the visualization of high-dimensional data [9]. Unlike other artificial neural networks, SOMs use neighborhood functions to generate the topological properties of the input vectors. Therefore, SOMs are really suitable for converting complex statistical relationships between high-dimensional data items into simple geometric relationships on a low-dimensional display. This is the season we choose the SOM to generate the trend vectors from the high-dimensional MDN structure for low-dimensional visualizations.

The training part of SOMs utilizes competitive learning. When a training example is fed to the network, its Euclidean distance to all weight vectors is computed. The neuron with weight vector most similar to the input is called the best matching unit (BMU). The weights of the BMU and neurons close to it in the SOM lattice are adjusted towards the input vector. The magnitude of the change decreases with time and with distance from the BMU.

During mapping, there will always be one single winning neuron: the neuron whose weight vector lies closest to the input vector. This can be simply determined by calculating the Euclidean distance between input vector and weight vector. In the case of ties a newer winning neuron will be chosen. After the mapping process is finished, the weighted neurons of the SOM then can be used as representations for all the vectors that are close to them. In other words, the trends of the neurons, which are the weights, can represent the general trends of all the nodes that are close to them. Therefore, all the neurons together are the trend of the network that contains all the nodes from the input vectors.

6 Experiments

The main purposes of the experiments are: 1) verify the basic part of the evaluation method in the workflow, 2) develop this method as comprehensive as possible. During the experiments section, we first extract the most essential and basic part from our workflow, which is the modified PageRank algorithm, in order to verify the correctness and usefulness of our workflow and the MDN structure. Then we gradually develop the workflow by adding machine learning method and considering more parameters for the weight system of the modified PageRank algorithm. Meanwhile, we also develop the MDN structure from a very simple form to a structure that represents a dynamic dataset changing through time.

6.1 Method Verification

The first experiment will be testing the basic part of the evaluation method, which is shown in Figure 4. To verify whether the result from the modified PageRank algorithm can be easily visualized as trend information for nodes in the target network, the full Facebook dataset will be used here. Because the goal here is verification, the MDN structure is also simplified in Figure 5. In the simplified example of MDN structure, the N_1 is a sub-network of N_2. In this particular case, we divided the whole Facebook dataset

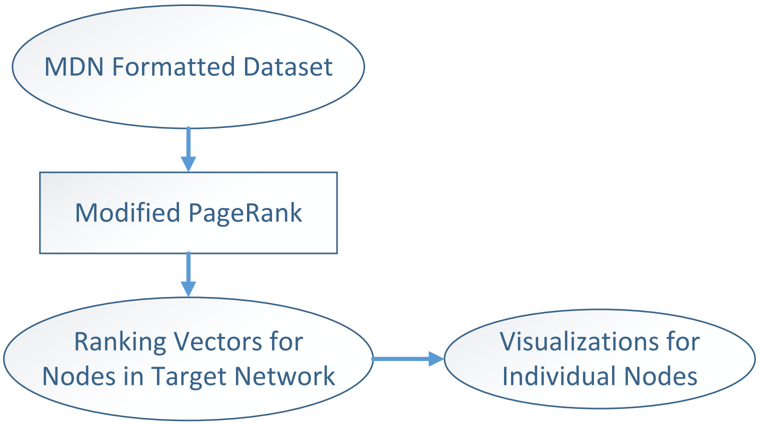

MDN Formatted Dataset
Modified PageRank
Ranking Vectors for Nodes in Target Network
Visualizations for Individual Nodes

Figure 4: The simplified workflow for verification

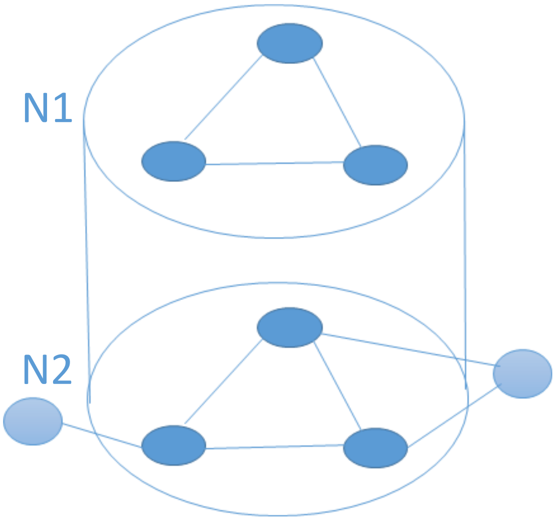


Figure 5: The simplified example of MDN structure for verification

into communities, individual communities are the candidates of N_1, the whole dataset is N_2. The length of ranking vectors generated from the modified PageRank algorithm will be 2, so we simplified the name of the two points in the ranking vectors as Local Ranking (for N_1) and Global Ranking (for N_2). The underlying data structure and a couple of example instances are shown in Table 2 and Table 3.

ID	Community	Global Ranking	Local Ranking
2703	9	0.00023	0.05779
3382	9	0.00022	0.05423

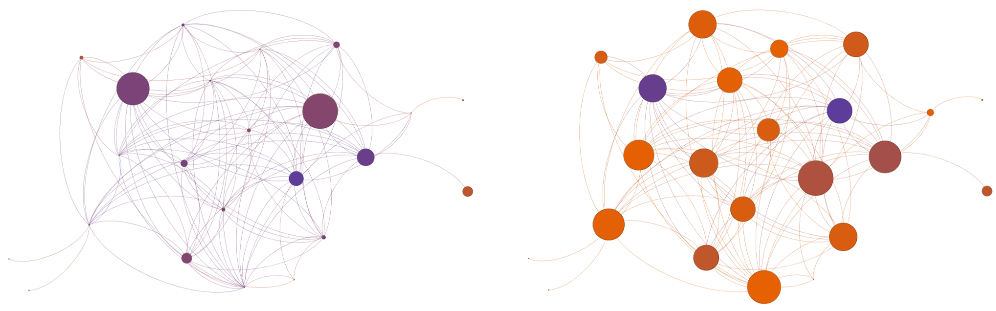
Table 2: Structure for nodes in the Facebook dataset

Source	Target
1684	3234
483	537

Table 3: Structure for edges in the Facebook dataset

Table 2 shows the structure for a single node in the Facebook dataset, including a node's ID. The Community ID is given by the community detection algorithm, the ranking score within the full dataset, and within the community. Table 3 has the structure of an edge from one node to another, edges in this dataset are undirected.

After using Modularity, the whole dataset, which has around 4,000 nodes, is divided into 16 communities. Each community contains from around 20 to 500 nodes. First, for illustration purposes, the visualization result of the smallest community, which is Community 9, is presented in Figure 6. Figure 6a shows each node's global ranking score by color, where the cooler the color is, the higher score it has, and local ranking score by size, the larger the higher. Likewise, Figure 6b show each node's local ranking score by color and the global ranking score by size. Combining these two visualizations, the different measurements and values for single nodes regarding global and local aspects can be easily found. It is easy to see that there are two nodes with significantly higher local ranking score than others. Compared to local ranking scores, global ranking scores are more on average within this community.

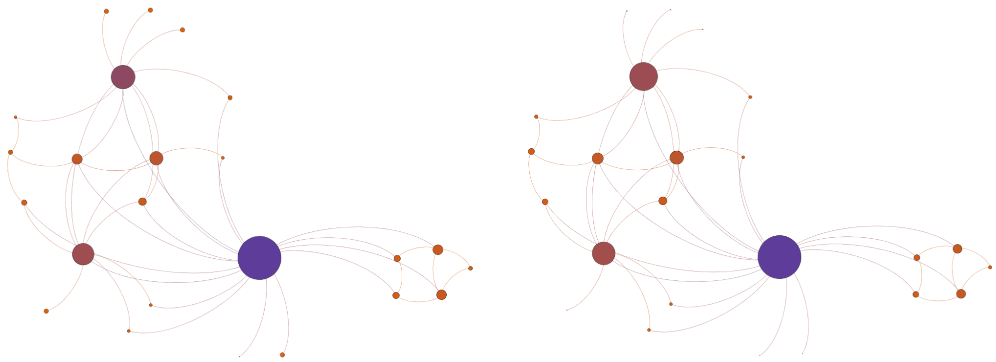


(a) size: local ranking, color: global ranking

(b) size: global ranking, color: local ranking

Figure 6: Example result of Facebook dataset generated with Gephi [22]

Because the Facebook dataset is relatively large, the second experiment will test if this method will work well on small networks. The dataset we chose for this part of our experiment is the famous Zachary karate club network [5]. The workflow and MDN structure configurations for the experiment are exactly the same as for the first experiments. After Modularity, the dataset is divided into 2 communities. The visualization result of the first sub-network is shown in Figure 7. Figure 7a shows each node's global ranking score by size, local score by color. Figure 7b shows the same information in the reverse way, i.e., size for local score and color for global score. Combining these two visualizations, we can see that the difference between global ranking scores and local ranking scores is very small if the number of communities is very low. And nodes at the edge of the community often have higher global ranking scores due to their "bridge" function.



(a) size: global ranking, color: local ranking (b) size: local ranking, color: global ranking

Figure 7: Karate dataset as small scale result

6.2 Method Development

The next step of the experiment is to develop the workflow to the full version, as well as the MDN structure. For this part the dataset HEP-TH is selected for the reason that it has the most specific meta details for every node and edge in our candidate pool so far. The other important advantage of HEP-TH dataset is that it is naturally categorized for the MDN structure by the year of publication. The MDN structure for the HEP-TH dataset is shown in Figure 8.

The original definition of an edge, which is the source paper cites the target paper, cannot serve this purpose well as the source paper and the target paper may not be published in the same year. Therefore, the new definition of the edge is provided here, which is the source paper and the target paper share the same references and/or the same authors. In this way, the nodes on both sides of the edge can be two papers published in the same year, which not only satisfies the current experiment goal, but also leaves room for further experiments. Note that the edges are undirected here. The weight of edges is also taken into account here because the sharing references and the sharing authors are not unique. The definition of the edge weight w for paper p_1 and p_2 is defined as follows:

$$w = \alpha \cdot rd(p_1, p_2) + \beta \cdot ad(p_1, p_2) \tag{4}$$

Here rd and ad, which are the shared reference distance and the shared author distance respectively, are defined as follows:

$$rd(p_1, p_2) = |R_1 \cap R_2| \tag{5}$$

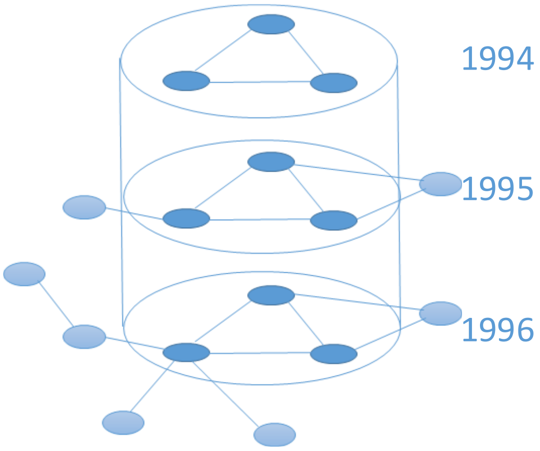


Figure 8: The example of MDN structure for HEP-TH dataset

$$ad(p_1, p_2) = \frac{|A_1 \cap A_2|}{|A_1 \cup A_2|} \tag{6}$$

Here R_1 and R_2 are the set of references for p_1 and p_2, respectively. Likewise, A_1 and A_2 are the sets of authors for p_1 and p_2, respectively. Finally, α and β are the coefficient parameters for rd and ad, respectively. The detailed value setting for α and β will be discussed later.

Therefore, the PageRank algorithm with edge weight will be:

$$PR_N(n) \leftarrow \frac{1-d}{|N|} + d\left(\frac{PR_N(n_1)}{W(n_1)} + \frac{PR_N(n_2)}{W(n_2)} + \cdots\right) \tag{7}$$

Where $W(n)$ is the sum of the weights of outbound links for the node n or the sum of the weights of links between node n and its neighbor nodes, depends on whether the links are directed or undirected. Please notice that the link weights only affect the PageRank value distribution of all nodes. The sum of PageRank value of all nodes still remains one. Likewise, the general decreasing trend of PagaRank value while the dataset getting larger still remains. The experiment below shows that our method will still eliminate that trend.

Because of taking multiple datasets (total 10 sub-datasets for HEP-TH) into account, as well as the link weights, the final data structure and a couple of example instances for dataset HEP-TH is shown in Table 4 and 5.

Table 4 shows the structure for a single node in the HEP-TH dataset, including a node's ID, and all the ranking values from the publishing year to the nth year. Table 5 shows the structure of the edge from one node to another, included the shared reference distance

ID	Ranking in Year 1	Ranking in Year 2	Ranking in ...	Ranking in Year n
9407001	7,50E-04	7,14E-04	...	6,81E-04
9410117	3,12E-04	2,62E-04	...	3,15E-04

Table 4: Structure for nodes in HEP-TH dataset

Source	Target	rd	ad	Weight
9401107	9401115	1	0.0	1
9401147	9412235	3	0.5	3

Table 5: Structure for edges in HEP-TH dataset

and the shared author distance. The edge represents the similarity of two nodes according to the weight.

The dataset HEP-TH is quite large. Before we actually go into it, we decided to create an artificial dataset as illustrated in Table 6 and Figure 9 to simulate the real HEP-TH dataset, in order to test the full version workflow without any possible influences coursed the dataset scale. The first two digits of the node ID are the publication year of this paper. The parameters α and β are set to $\alpha = 1$ and $\beta = 0$ in Equation 4 for the purpose of simplifying the result. In this artificial dataset, all the nodes are added to the network in four years.

Source	Target	Rd	Ad	Weight
0000	0001	3	0	3
0000	0002	3	0	3
0001	0002	3	0	3
0100	0001	3	0	3
0101	0001	3	0	3
0102	0001	3	0	3
0200	0000	3	0	3
0201	0000	3	0	3
0300	0001	3	0	3
0301	0001	3	0	3

Table 6: Edges in artificial HEP-TH dataset

The developing progress from Year 0 to Year 2 is shown in Figure 10, and the final stage is shown in Figure 9.

When looking into the ranking score for one particular node in more than two different groups, like our example here, the disadvantage of the original Pagerank algorithm, which mentioned in Section 5.2, will cause too much interference. Figure 11 shows the developing curve for the sample dataset from Year 0 to Year 3 without normalization. It

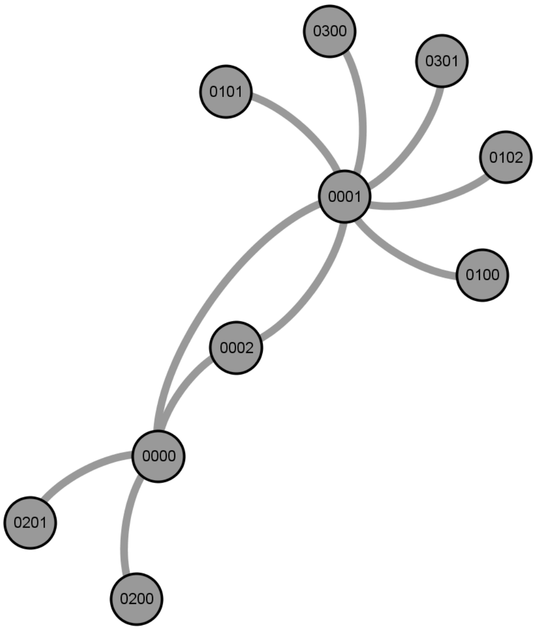
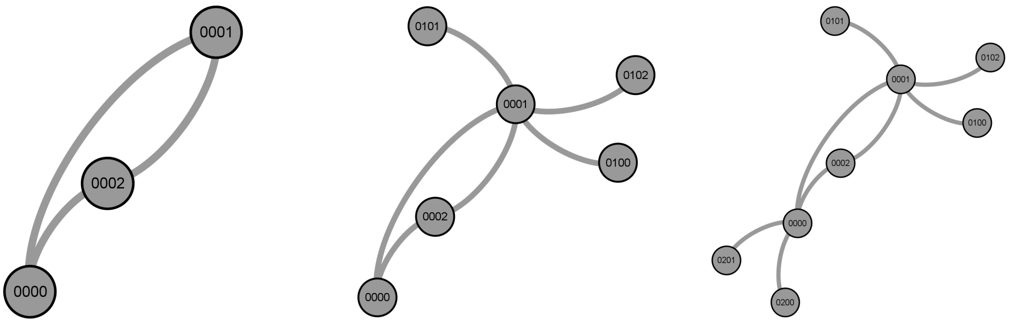


Figure 9: Visualization for the artificial dataset at Year 3

can be easily noticed that the curves of all three nodes are moving downwards. However, developing curves are not the direct reflection of how the value of each node changed during four years. It is already difficult for observation in such a simple dataset, not to mention that the actual datasets always have different weights of links. Therefore, the step of normalization is necessary for finding the trend for the nodes in multiple dimensions.

Here we focus on the first three nodes, which are published in Year 0. If our method works correctly, the three nodes should have the same ranking value at Year 0, and then present different developing trends during the next four years. Node 0000 should get higher ranking value in Year 2, and remains stable at other years, because there are 2 new nodes building connections with Node 0000 in Year 2. Node 0001 should increase all the time except Year 2 because there are new connections every year except in Year 2, and Node 0002 should remain unchanged all the time. After we went through the modified PageRank algorithm, the result, which is shown as Figure 12, is exactly the same as we expected.

The vertical axis (NPr) is the ranking score after generated from our modified PageRank algorithm (Equation 3). The decreasing of the curve of the Node N will only have happened when the actual value of the links of the Node N decreasing. This situation will appear if some nodes with high link weight are added to the network of Node N. Notice that compared to the normal PageRank, the ranking score here can be higher than 1 due to the normalization step. Therefore, our method not only works at “geographic



(a) Year 0 (b) Year 1 (c) Year 2

Figure 10: Developing progress from Year 0 to Year 2

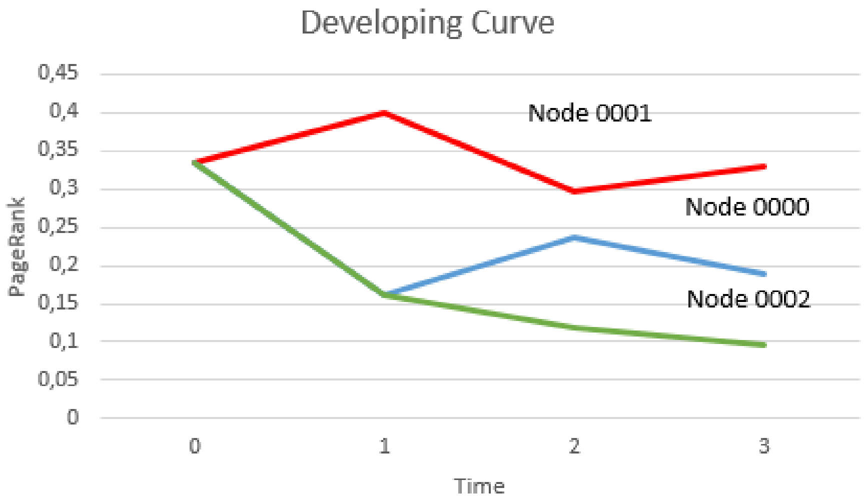


Figure 11: Developing curves for the artificial dataset from Year 0 to Year 3 without normalization

dimensions", but also goes along with time.

For the experiment on the real HEP-TH dataset, we selected the publications from the year 1994 as basic dataset. Some data instances are listed in Table 7. The first two digits of node ID here are the year, followed by two digits representing the month. Before we go through the workflow, we are also interested in the positioning information for the dataset like the artificial dataset. The 100 nodes with most degrees from the target network (1994) are processed by Forced Atlas Algorithm [21] for positioning visualization. The positioning information for 1994, 1998, and 2003 is shown in Figure 13.

Some statistic data, which also can show how the positioning of these 100 nodes developed through time is shown as Table 8. In order to get the above statistic information, we

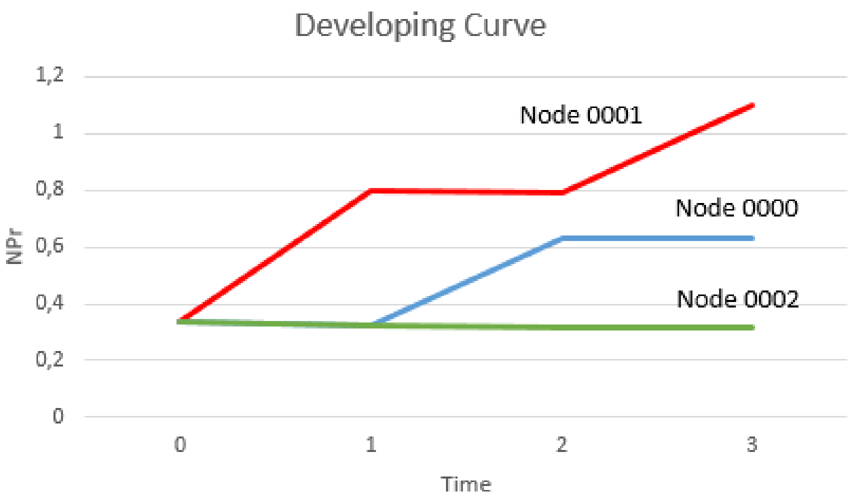


Figure 12: Developing curves for the artificial dataset from Year 0 to Year 3

Source	Target	Rd	Ad	Weight
9403018	9411020	3	0.00	3
9403026	9405136	2	0.33	2
9403026	9408069	2	0.50	2
9404041	9411053	7	0.60	7
9404042	9411181	1	0.00	1
9404067	9408036	4	0.75	4

Table 7: Example edges in the Year 1994 HEP-TH dataset

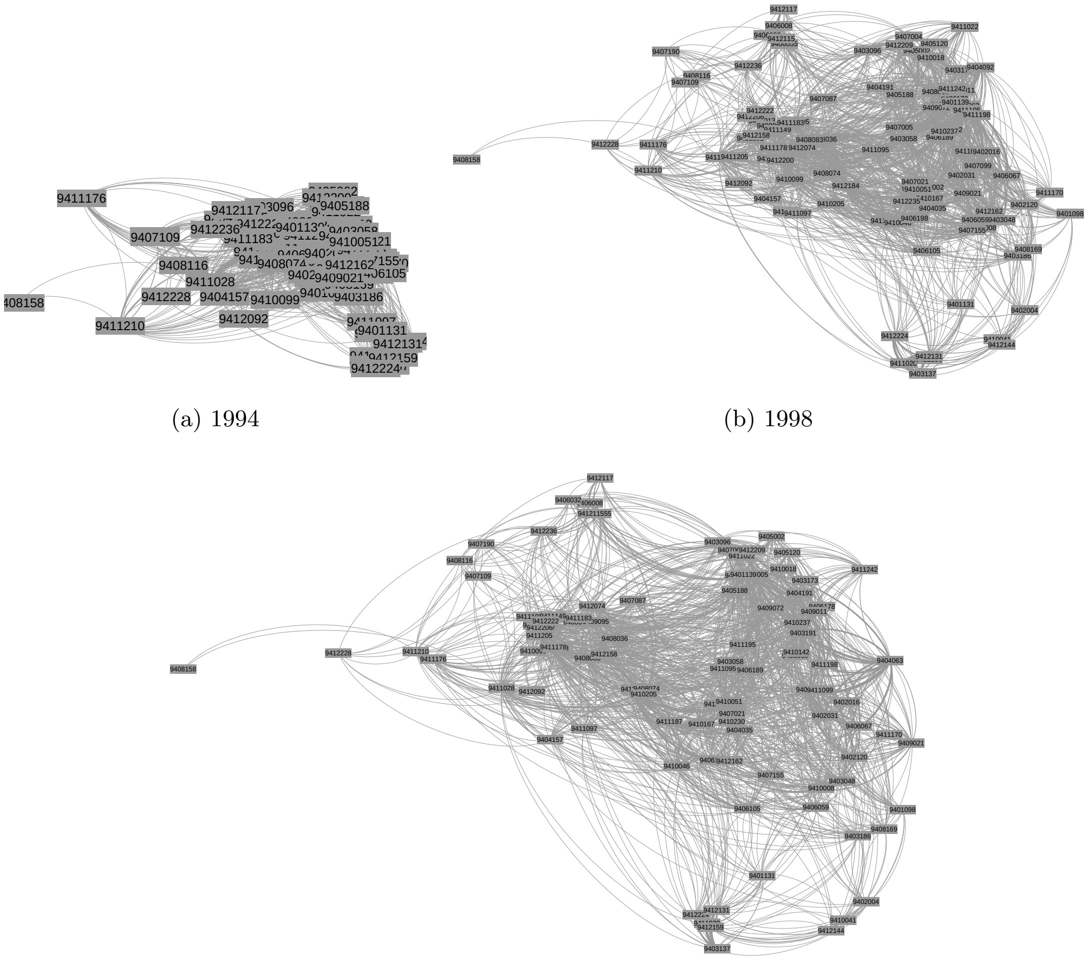
By Pixels	94To98	98To03	94To03
Avg.	674,8194381	428,484735	990,3777401
Min.	107,105713	7,710565105	116,3690459
Max.	1713,658102	1213,778808	2273,875986

Table 8: Node positions moving amount during time

extracted the coordination data for all nodes from the the visualization result, and computed the Euclidean distance between different year's positions for the same node.

The positioning information provides a straightforward way to find out the interesting nodes which could be the potential research target. This information can be put together with trend information of individual nodes for reference and comparison.

Next, the nodes in the target network, which are the papers in Year 1994, will be processed to experiment. The start configuration of this experiment is the same as before, which has again $\alpha = 1$ and $\beta = 0$. The result of some instances are shown in Figure 14.



(a) 1994

(b) 1998

(c) 2003

Figure 13: Positioning information for HEP-TH dataset

We put the node's developing curve together with its positioning information in Figure 15 and Table 9.

By Pixels	94To98	98To03	94To03
908074	215,9175534	305,7927479	116,3690459

Table 9: Node positions moving amount for paper 9408074

The visualization result becomes more comprehensive because the positioning information can also reflect the developing trend of nodes by number.

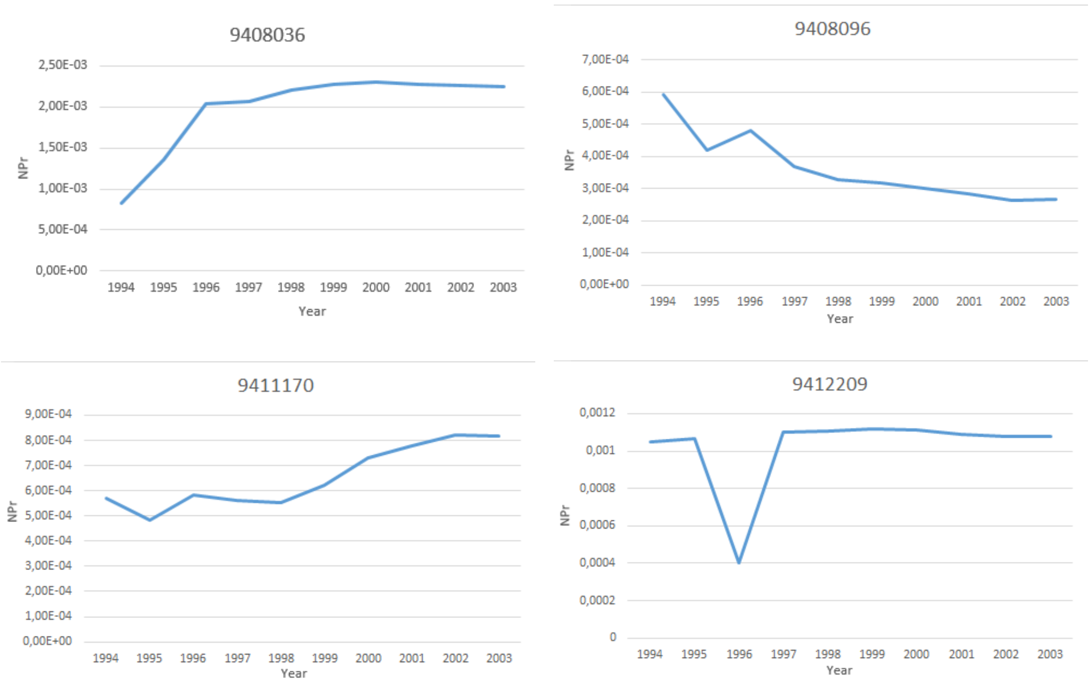


Figure 14: Developing progress from the Year 1994 to the Year 2003 for some instrances

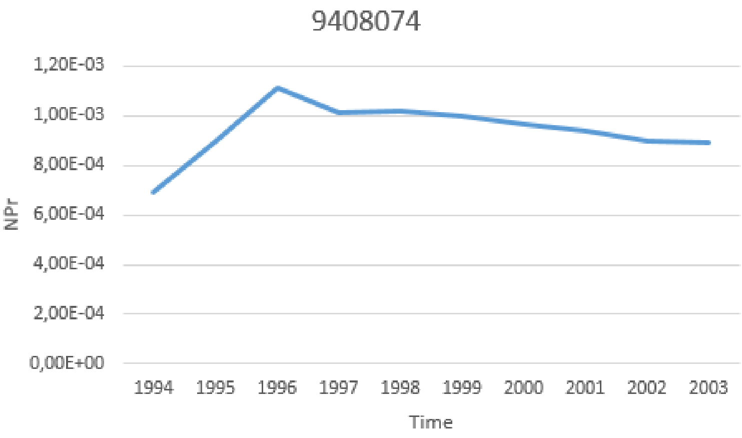


Figure 15: Developing progress for paper 9408074

To figure out the general trends of the dataset from the Year 1994 to the Year 2003, we clustered all instances in our dataset by means of a Self-Organizing Map, which has two parameters w and h representing width and height, respectively, needed to be configured in order to decide the size of the SOM. To do so, we use the Modularity Algorithm to check how many communities there are within our dataset. The result is shown in Figure

16. The number of communities is 164, most of them having very few members. This

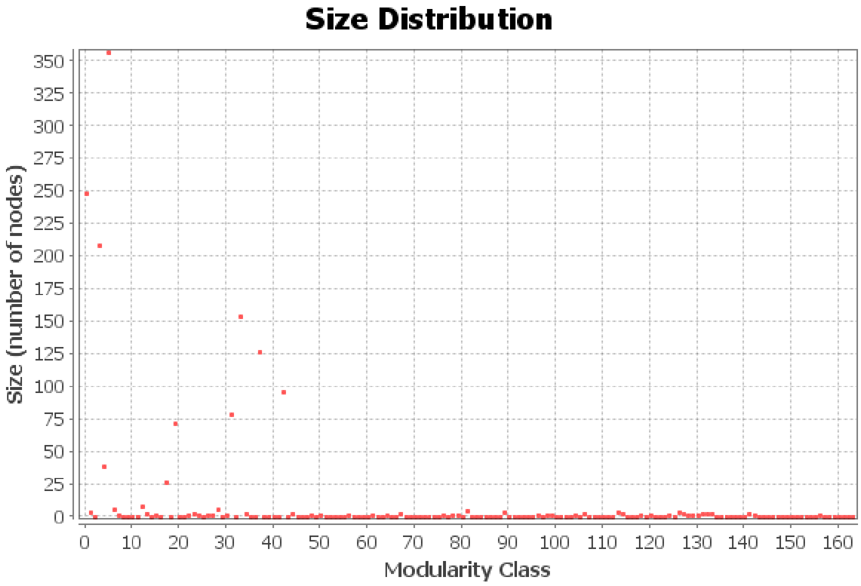


Figure 16: Modularity result for the Year 1994 HEP-TH dataset

situation is due to the natural attributes of the dataset itself: the links are the shared references. The nodes are only the papers published in the Year 1994, most of them only have very few shared references with a small amount of other papers. But we can still find out that there are about 10 communities containing more than 25 nodes. Therefore, to be relatively more representive, we configure the size of the SOM to 3 × 3. The result generated from the SOM shows us 9 clusters with different sizes and trends, and the details of these 9 clusters are shown in Table 10 and Figure 17. Combining Table 10 and

Communities	Nr. of Nodes
0	125 (8%)
1	486 (30%)
2	449 (27%)
3	45 (3%)
4	246 (15%)
5	173 (11%)
6	10 (1%)
7	23 (1%)
8	90 (5%)

Table 10: The result generated from the SOM

Figure 17, we can easily find out several general trends for the Year 1994 dataset.

First, the majority (80%) of the papers have very low (less than 0.001) *NPr* values

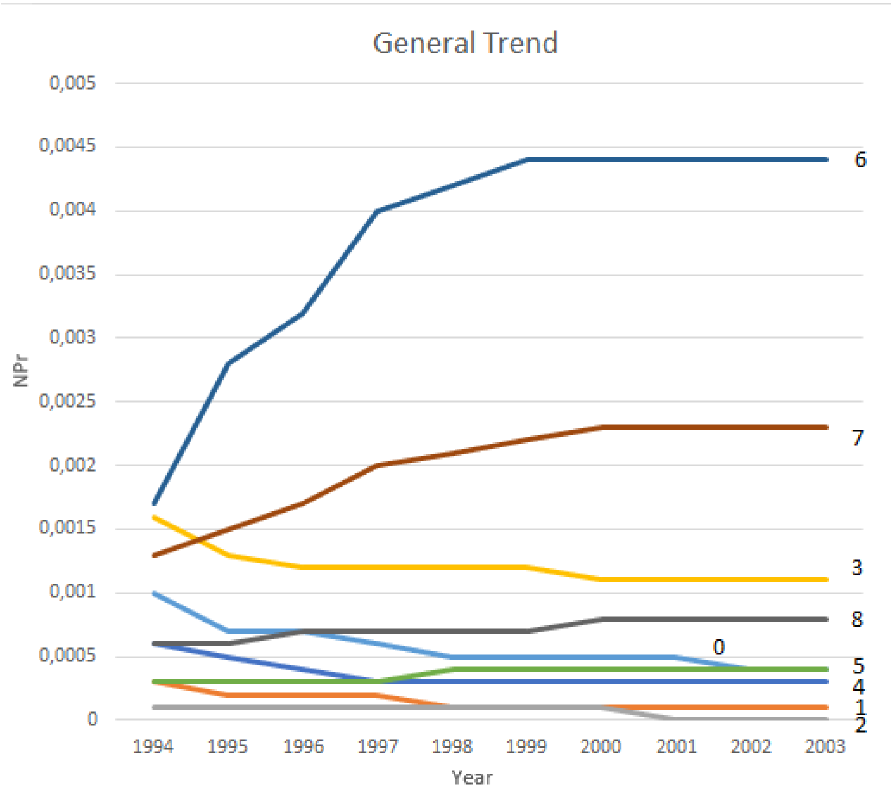


Figure 17: The general trends generated by the SOM

(Community 0, 1, 2, and 4), and the values generally decrease through time, although different communities have different shapes of the decreasing curve, such as 30% of the papers (Community 1) decrease their values at an early stage and remain stable at the rest, 27% of the papers (Community 2) lose their value a couple of years later after their publication date.

Second, 16% of the papers (Community 5 and 8) increase their values during time, although overall they still have a relatively low value.

Third, only 4% of the papers (Community 3, 6 and 7) have a high value since they are published, but more than half of them (Community 3) decrease their values during 10 years, whereas only 33 out of 1647 papers (Community 6 and 7) show pure positive trends.

Last but not least, no matter how the trend curve developed in the beginning, after 5 to 8 years, the value of all the papers tend to be stable. This means very few new papers are joined into the similarity group of these more than 5 years old papers.

6.3 Variation of Link Weight

Until this moment, we only considered the shared references as the standard for the similarity of two papers, as $\beta = 0$ in Equation 4. Although the experiment result is acceptable, we are wondering what information we can extract from putting more variety to link weight.

The major problem of taking shared authors into consideration is how the shared authors contribute to the similarity between two papers. It is not certain that two papers having more shared authors means that there is more similarity between them or the other way around. A very high number of shared authors of two papers can either mean that these two papers are very similar, or that this group of authors are in the same community, such as colleagues in the same research team.

The decision between assigning positive effect or negative effect of shared authors to the similarity between two papers is hard to make. However, combining shared authors and shared references together can simplify this problem: if two papers already have many shared references, that means these two papers are already very similar, then having more shared authors means more similarity; if two papers have very few shared references, that means they do not have much similarity between each other, then it could be other reasons like we mentioned before than similarity causes the high number of shared authors. Therefore, we developed a way to determine β in Equation 4:

$$\beta = \begin{cases} +1 & \text{if } rd(p_1, p_2) \geq avg(rd) \\ -1 & \text{if } rd(p_1, p_2) < avg(rd) \end{cases}$$

Here $avg(rd)$ is the number of the average shared references for the nodes in the basic dataset. In this case, $avg(rd)$ is the number of average shared references for papers only in Year 1994, which is 1.76, so if there are 2 or more than 2 shared references between two papers, the effect of shared authors is considered to be positive, otherwise, it is negative.

There is one exceptional circumstance, which is $rd = 1$ and $ad = 1$ in Equation 4. Because the edge weight cannot be 0, a small value (0.1) will be assigned to the edge weight in this situation.

After re-calculating the edge weights, we did the same process as in the section above. Figure 18 and Figure 19 show some examples that can be compared with the result without taking shared authors into account. The curve for paper 9405182 (Figure 18) shows a more dynamic result while the general trend remains the same. However, the trend for paper 9408096 (Figure 19) changed completely. Due to our normalization method, the reason for the trend curve of paper 9408096 changing from downwards to upwards is that the authors for paper 9408096 wrote more papers shared with other authors during ten years, although the influence of paper 9408096 itself decreased during time.

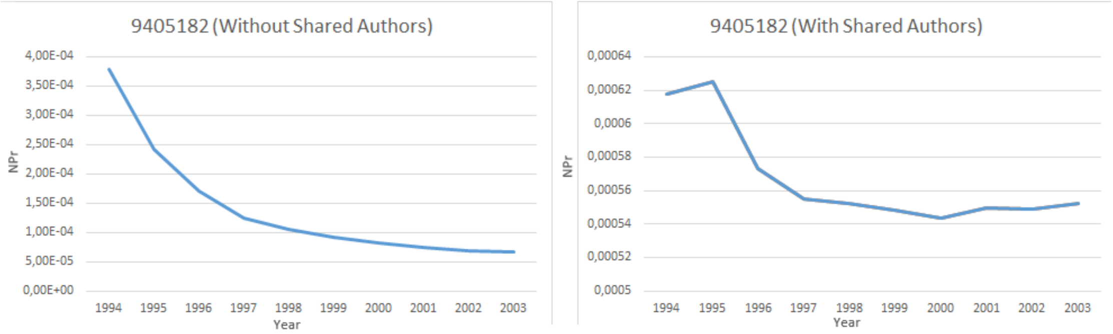


Figure 18: Result Comparison for paper 9405182

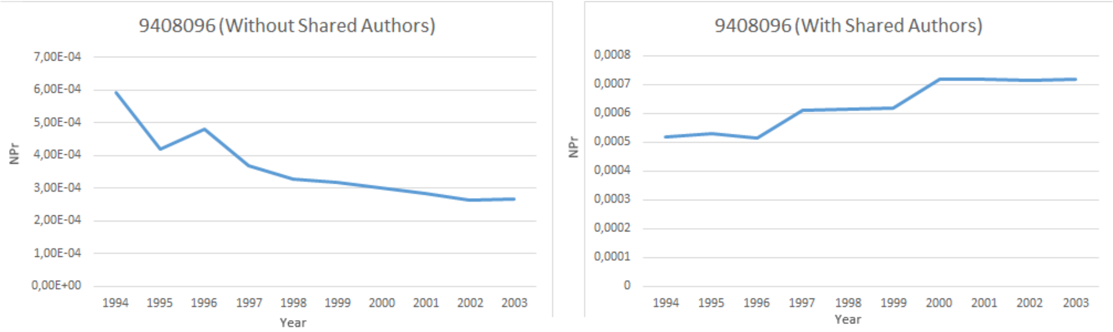


Figure 19: Result Comparison for paper 9408096

These two situations can basically represent the result if we compare all the trend curves between these two methods. The latter occurs more often than the former. The reason of course is that the parameter of shared authors has more effect on some papers than others, which means the authors of the papers that changed their curves from downwards to upwards (like paper 9408096), had more publications than the authors of the paper that remains going downwards (like paper 9405182). Although it is hard to tell which method is better, the result with shared authors does provide us more variety and more detailed information.

Some readers may notice that the two example curves above both go downwards in the result without shared authors. That is because we can barely find the sample that shows the papers have upwards trend curves in the result without shared authors changed their trend curves to downwards in the result with shared authors. This circumstance reflects that the authors who wrote some high value papers tend to write more papers in the future. The SOM result verified our assumption as can be seen in Figure 20 while the community distribution (Table 11) remains almost the same. As we mentioned before, the reason why the trend curve with shared authors is more smooth and generally increasing is: most papers get less and less shared references during years, but their authors will keep writing new papers with a smooth speed (according to our result).

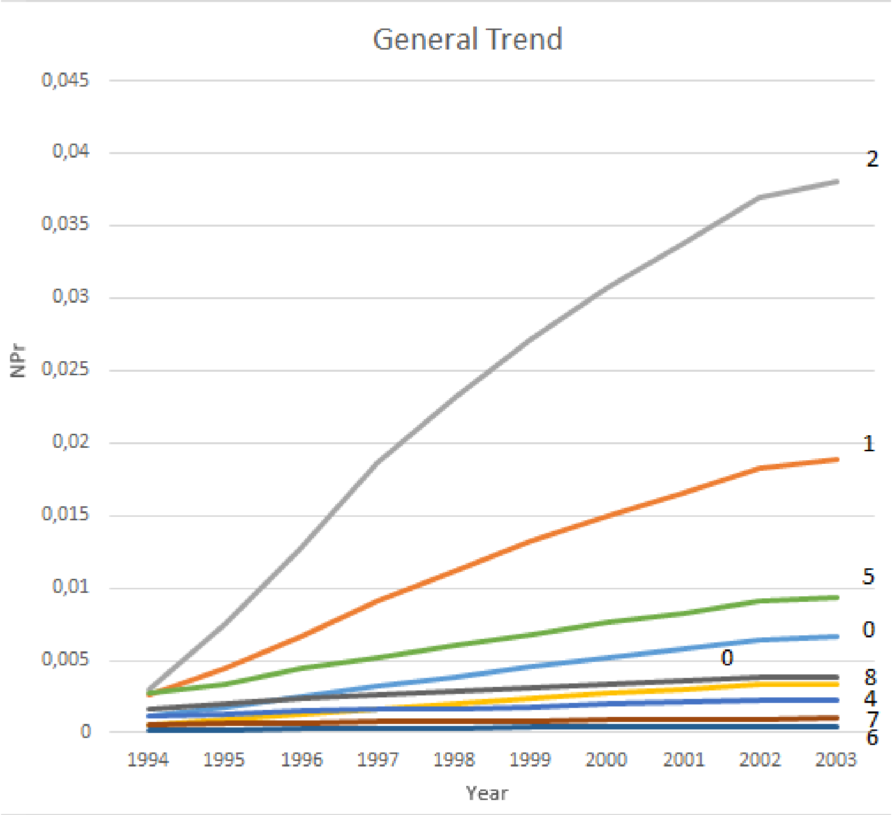


Figure 20: The general trends with shared authors generated by the SOM

Communities	Nr. of Nodes
0	93 (6%)
1	27 (2%)
2	11 (1%)
3	159 (10%)
4	230 (14%)
5	48 (3%)
6	491 (30%)
7	459 (28%)
8	129 (8%)

Table 11: The SOM result with shared authors

Likewise, the small amount of authors with popular papers (Group 6, 7 in Table 10) also writing more new papers during time (Group 1, 2 in Table 11).

7 Conclusion and Future Work

In conclusion, a dataset structure called Multiple Dynamic (MDN) structure is created in this thesis as the fundamental platform of our research. Based on this structure, we combined the PageRank algorithm, which is modified with normalization for the MDN structure, with Self Organizing Maps and the Modularity Algorithm to a complete workflow in order to evaluate the influential trend for nodes contained in multiple dynamic networks.

As experiment results, multiple types of visualizations were generated for influential trend information of the citation network through time, including positioning changing information of nodes, trend of individual nodes through time, as well as trend information of the network itself. Moreover, the function and effectiveness of the workflow were proven with both large scale and small scale datasets. The limitation of this evaluation method is that the input dataset for this method has to be either naturally matched by the MDN structure, or can be processed to the MDN structure.

For future work, there are at least two possible directions. First, we can examine the advantage of the SOM's ability to predict the future trend of both existing nodes and new nodes connected into the citation network. Second, we want to develop a general sub-workflow for processing the raw dataset to the MDN structure, so that the entire workflow can be automated.

References

[1] Trudeau, J. Richard, *Introduction to Graph Theory.* New York: Dover Pub. p. 19. ISBN 978–0–486–67870–2, 1993

[2] J. Leskovec, *Stanford Large Network Dataset Collection*, http://snap.stanford.edu/data/index.html [accessed May. 19, 2015].

[3] Facebook. http://www.facebook.com [accessed May. 19, 2015].

[4] Twitter. http://www.twitter.com [accessed May. 19, 2015].

[5] W. Zachary, *Zachary karate club network dataset*, http://konect.uni-koblenz.de/networks/ucidata-zachary [accessed May. 19, 2015].

[6] E-print arXiv. http://arxiv.org/ [accessed May. 19, 2015].

[7] Wikipedia, http://en.wikipedia.org/ [accessed May. 19, 2015].

[8] M. E. J. Newman, *Modularity and community structure in networks*, In Proceedings of the National Academy of Sciences of the United States of America 103 (23): pp. 8577–8696, 2006.

[9] S. Haykin, *Self–organizing Maps Neural Networks–A Comprehensive Foundation*, 2nd ed., Prentice-Hall, 1999.

[10] V. D. Blondel, J. L. Guillaume, R. Lambiotte, E. Lefebvre, *Fast unfolding of communities in large networks*, In Journal of Statistical Mechanics: Theory and Experiment (10), P1000, 2008.

[11] S. Brin, L. Page, *The Anatomy of a Large-Scale Hypertextual Web Search Engine*, In Proceedings of the Seventh International Conference on the World Wide Web (WWW): pp. 107–117, 1998.

[12] Ding, Ying, E. Yan, A. Frazho, and J. Caverlee, *PageRank for ranking authors in co–citation networks*, Journal of the American Society for Information Science and Technology 60, no. 11: 2229–2243. 2009

[13] Ding, Ying. *Applying weighted PageRank to author citation networks*, Journal of the American Society for Information Science and Technology 62, no. 2: 236-245. 2011

[14] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi, *Measuring user influence in twitter: The million follower fallacy*, In 4th International AAAI Conference on Weblogs and Social Media (ICWSM) pp. 14–1–8. 2010

[15] M. Luiten, W.A. Kosters, and F.W. Takes, *Topical Influence on Twitter: A Feature Construction Approach*, In Proceedings of 24th Benelux Conference on Artificial Intelligence (BNAIC'12); pp. 139–146, 2012.

[16] F.W. Takes , W.A. Kosters, *Determining the Diameter of Small World Networks*, In Proceedings of 20th ACM Conference on Information and Knowledge Management (CIKM 2011): pp. 1191–1196, 2011.

[17] F.W. Takes, W.A. Kosters, *The Difficulty of Path Traversal in Information Networks*, In Proceedings of 4th International Conference on Knowledge Discovery and Information Retrieval (KDIR '12): 138–144, 2012.

[18] P. Kazienko, K. Musial, and A. Zgrzywa, *Evaluation of Node Position Based on Mutual Interaction in Social Network of Internet Users*, II Krajowa Konferencja Naukowa Technologie Przetwarzania Danych, (TPD 2007): 265–276, 2007.

[19] J. Weng, E. P. Lim, J. Jiang, and Q. He, *Twitterrank: Finding Topic-Sensitive Influential Twitterers*, In Proceedings of the 3rd ACM International Conference on Web Search and Data Mining: pp. 261–270, ACM, 2010

[20] S. Wu, J. M. Hofman, W. A. Mason, and D. J. Watts, *Who Says What to Whom on Twitter*, In Proceedings of the 20th International Conference on World Wide Web: 705–714, ACM, 2011

[21] M. Jacomy, T. Venturini, S. Heymann, M. Bastian, *ForceAtlas2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software*, PLoS ONE 9(6): e98679. doi:10.1371/journal.pone.0098679, 2014

[22] The Open Graph Viz Platform. https://gephi.github.io/ [accessed May. 19, 2015].