T.C. DOĞUŞ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
COMPUTER AND INFORMATION SCIENCES DEPARTMENT

AN APPLICATION OF COMMUNITY DISCOVERY IN ACADEMICAL SOCIAL NETWORKS

M.S THESIS

Enis ARSLAN
200991004

Thesis Advisor:
Prof. Dr. Selim AKYOKUŞ

JANUARY 2013
ISTANBUL

T.C. DOĞUŞ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
COMPUTER AND INFORMATION SCIENCES DEPARTMENT

AN APPLICATION OF COMMUNITY DISCOVERY IN ACADEMICAL SOCIAL NETWORKS

M.S THESIS

Enis ARSLAN
200991004

Thesis Advisor:
Prof. Dr. Selim AKYOKUŞ

JANUARY 2013
ISTANBUL

Doğuş Üniversitesi Kütüphanesi
0007726

DOĞUŞ ÜNİVERSİTESİ
KÜTÜPHANESİ

T.C. DOĞUŞ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
COMPUTER AND INFORMATION SCIENCES DEPARTMENT

AN APPLICATION OF COMMUNITY DISCOVERY IN ACADEMICAL SOCIAL NETWORKS

M.S THESIS

Enis ARSLAN
200991004

Thesis Advisor:
Prof. Dr. Selim AKYOKUŞ

JANUARY 2013
ISTANBUL

PREFACE

In my thesis, Community Detection algorithms and methods that discover the communities in the social networks are applied by two different methods on two different datasets. Two datasets: DBLP and Arxiv citation network datasets are used in this thesis. Detected groups and communities are discovered by using Main path analysis and k-core community discovery process.

Istanbul, January 2013 Enis ARSLAN

ABSTRACT

The objective of this thesis is to discover social communities in a social network using different social network community discovery methods that utilizes metrics and structures like degree, clustering coefficient, k-cores, weak and strong components. In this study we have used two different datasets: DBLP and Arxiv High-energy physics theory citation network.

Two Social Network Analysis tools are used in this thesis: Pajek and Gephi. In order to use Pajek and Gephi, DBLP dataset is converted by developing a new conversion and refinement framework. After dataset conversion, we have used Pajek tool to discover communities by applying several clustering metrics to the social networks. Additionally, Gephi tool is used for supporting the analysis of discovering communities by using extended metrics. Gephi tool enables visualization of the results graphically and gives the reports of the analyses.

At the end of the analyses, we have obtained several reports and graphs that show triads and skeleton structure of the communities in the networks. These reports and graphs give social communities and the leaders of networks and several characteristics of these communities.

ÖZET

Bu tezin amacı, degree, clustering coefficient, k-cores, weak, strong components gibi çeşitli sosyal ağ topluluk ölçü ve yapılarını kullanarak bir sosyal ağ'daki sosyal toplulukların keşfedilmesidir. Bu çalışmada iki farklı veri seti kullanılmıştır: DBLP ve Arxiv High-energy physics theory citation ağı.

Bu tezde iki Sosyal Ağ Analizi programı kullanılmıştır: Pajek ve Gephi. Pajek ve Gephi'yi kullanabilmek için yeni bir framework tasarlanarak, DBLP veri kümesi çeşitli rafine etme ve düzenleme işlemine tabi tutulmuştur. Veri kümesi düzenlemelerinden sonra, Pajek programı birçok kümeleme metriklerini sosyal ağ'lara uygulayarak toplulukları keşfetmek için kullanılmıştır. Bunlara ek olarak, Gephi programı ile ilave metrikleri kullanarak yapılan analiz desteklenmiştir. Gephi programı ile sonuçlar grafiksel olarak görselleştirilmiş ve analiz raporları hazırlanmıştır.

Analizin sonunda, sosyal ağlardaki sosyal toplulukların üçlü topluluk ve iskelet yapılarını gösteren çeşitli rapor ve grafikler elde edilmiştir. Bu raporlar ve grafikler sosyal ağlardaki sosyal toplulukları ve liderlerini, ve birçok topluluk karakterini göstermektedir.

ACKNOWLEDGEMENTS

I would like to express my deep appreciation and gratitude to my advisor Prof. Dr. Selim Akyokuş for his great guidance, support and encouragement he provided to me during my thesis study.

This thesis is dedicated to my parents, for their love, endless support and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

SNA	Social Network Analysis
MCL	Markov Clustering Algorithm
KL Algorithm	Kernighan Lin Algorithm
SPC	Search Path Count

1. INTRODUCTION

A social network is a social structure made up of individuals and organizations that form specific groups. Social networks can be an example of collaboration of colleagues in an organization or communities like Facebook, LinkedIn, mobile gaming communities. Social communication inside a social network can form a graph where the members are the nodes and communication values are the edges. Social Network graphs are dynamic structures where nodes can be added with new subscriptions and can be deleted with sign offs (Dasgupta et al., 2008).

Social network analysis (SNA) is the methodical analysis of social networks that maps and measures the relationships and flows between individuals, groups, organizations, computers, and other connected entities. There are lots of new concepts, terms and metrics used in social networks analysis like graphs, Paths, Components, Cores and Cliques, Clustering Coefficient, Transitivity, Centrality. In the first part of thesis, these concepts and terms are introduced and discussed.

In this thesis, it is aimed to discover social communities in a social network using different social network community discovery methods that utilizes metrics and structures like degree, clustering coefficient, k-cores, weak and strong components. There are several community discovery algorithms and metrics used in community discovery. Some of the community discovery algorithms are described in the second part of the thesis.

Community discovery in social networks can lead to applications in use of

- Link spamming
- Abnormal social groups detections
- Network Intrusion Detection
- Discovering network value for viral and targeted marketing

- Churn Prediction (Aggarwal C. C., 2011)

We have used two Social Network Analysis tools: Pajek and Gephi and two datasets: DBLP and Arxiv High-energy physics theory citation network.

DBLP dataset is converted by developing a new conversion and refinement framework. After dataset conversion, we have used Pajek tool to discover communities by applying several clustering metrics to the social network. Additionally, Gephi tool is used for discovering communities by using extended metrics. Gephi tool enables visualization of the results graphically and gives the reports of the analyses.

This thesis is organized as follows: Chapter 2 provides an introduction to network theory and the important network metrics. Chapter 3 describes structures, graph partitioning and community discovery algorithms used in social network analysis. Chapter 4 gives a report of discovered communities and leaders obtained from SNA datasets using Pajek and Gephi tools.

2. STUDY OF NETWORKS

2.1. Network Theory

In mathematical means, a network is a graph composed by collection of vertices connected by edges. Generally, n is the number of vertices and m is the number of edges. Some examples of the networks of different types are listed below in the Table 2.1.

Network	Vertex	Edge
Internet	Computer or router	Cable or wireless data connection
World Wide web	Web page	Hyperlink
Citation network	Article, patent, or legal case	Citation
Power grid	Generating station or substation	Transmission line
Friendship network	Person	Friendship
Metabolic network	Metabolite	Metabolic reaction
Neural network	Neuron	Synapse
Food web	Species	Predation

Table 2.1 Network Types
(Newman, M.E.J., 2011)

A network can be represented as an adjacency matrix $A[i,j]$ where: $A[i,j]=1$ if there is an edge between nodes i and j ; 0 otherwise

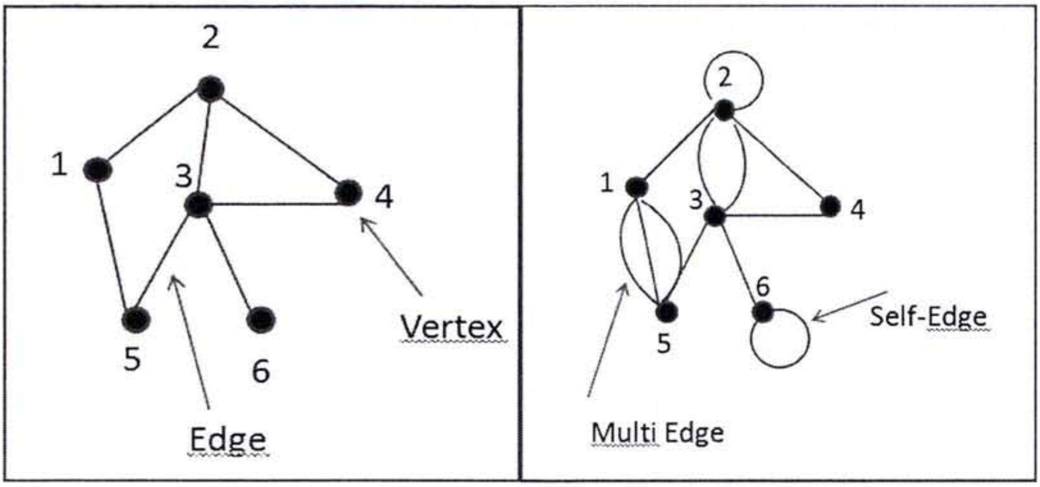


Figure 2.1 Simple graph and Multigraph
(Newman, M.E.J., 2011)

A simple graph is represented in Figure 2.1 at the left and the one at the right represents a multigraph with multiedges and self-edges.

Adjacency matrix for Figure 2.1 (left) is:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$ Eq.1 (Newman, M.E.J., 2011)

Note that it is symmetric because if there is an edge between i and j then there is an edge between j and i and diagonal matrix elements are zero. Adjacency matrix for Figure 2.1 (right) is :

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 3 & 0 \\ 1 & 2 & 2 & 1 & 0 & 0 \\ 0 & 2 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 3 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 2 \end{bmatrix}$$ Eq.2 (Newman, M.E.J., 2011)

A double edge between vertices i and j will be represented by 2 and a self-edge from edge i to i will be represented by the value of 2 in the diagonal because these edges have two ends.

Another representation of a network is adjacency list. In an adjacency list representation, a list of vertices adjacent to a vertex is stored on a list. An adjacency list is actually not just a single list, but a set of lists one for each vertex i. An adjacency list can be stored in series of integer one for each vertex or as a two dimensional array with one row for each vertex. Assuming a graph with m edges, storage of 2m integers is needed for an adjacency list. For example where n= 10,000 (n for vertices) and m=100,000 (m for edges) for integer of 4 bytes, if adjacency list is used 800 KB is needed where 400 MB storage is needed for an adjacency matrix (Newman, M.E.J., 2011).

2.1.1. Paths

Paths are the consecutive vertices connected by edges, in layman's terms a path is a route across the network that runs from vertex to vertex along the edges of a network. Paths can be in directed and undirected networks for the exceptional case that in directed paths they must follow the directions of the edges. Some paths can intersect itself by crossing the previous visited vertices. The path that does not intersect itself is called self-avoiding paths. Geodesic and Hamiltonian paths are examples of such paths. The length of a path is the number of edges traversed in the route of the path. A simple path for a directed path is shown in Figure 2.2

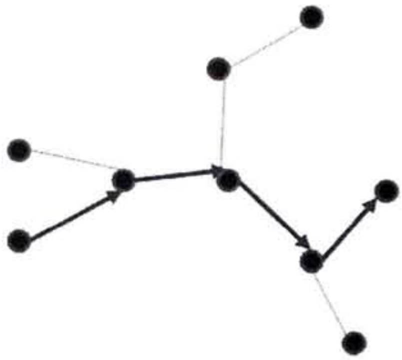


Figure 2.2 Path
(Newman, M.E.J., 2011)

A geodesic path is the shortest path between two vertices and they are self-avoiding. The length of a geodesic path is called the shortest distance or geodesic distance. A pair of vertices may have equal size geodesic paths. The diameter of a graph is the longest geodesic path between any of two of the vertices.

An Eulerian path is the path that passes each edge at least once. A Hamiltonian path is a path that passes each vertex at least once. An Eulerian path need not be self-avoiding because there may multi edges between any of the two vertices. As an example of an Eulerian path the people are very interested in the riddle Königsberg (Kaliningrad) problem in 1736. There are two islands and seven bridges in the middle of the river. The problem is starting from any point how to pass all bridges exactly once in a route. Euler has worked in this problem and he proved that there is no solution for this problem. In his opinion since any Eulerian path must both enter and leave every vertex, except the first and last, there can be at most two odd degreed vertices since four vertices have odd degree for Königsberg problem depicted in Figure 2.3 .

Figure 2.3 Königsberg Problem
(Newman, M.E.J., 2011)

Eulerian and Hamiltonian paths are applied in job sequencing, parallel programming and garbage collection in computer science.

2.1.2. Components

A component is the subgroup of vertices where there is at least one connection between each and no connection between subgroups. In Figure 2.4 there is a network with two components. A network of this kind is said to be disconnected while it is said to be connected if there is at least one path between them. A single vertex which has no connection with others is said to be a single component of size one.

Figure 2.4 Component
(Newman, M.E.J., 2011)

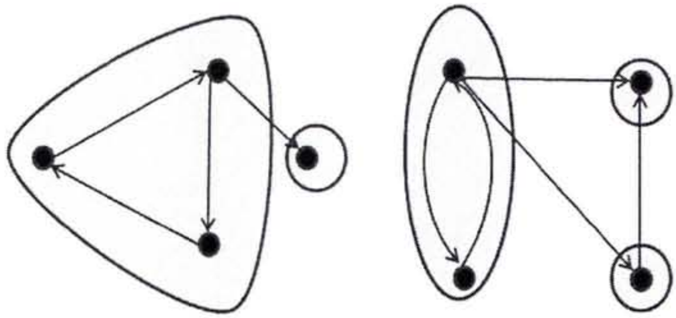


Figure 2.5 Weakly/Strongly connected components
(Newman, M.E.J., 2011)

For the Figure 2.5, if we ignore the directions of the edges, there are two components each with four vertices. These are weakly connected components. Two vertices are in the same weakly connected component if they are connected by one or more paths through the network. There are five strongly connected components in the Figure 2.6(shaded). In other words, a strongly connected component is a maximal subset of vertices such that there is a directed path in both directions between every pair in the subset. A strong connected component with more than one vertex must have at least one cycle.

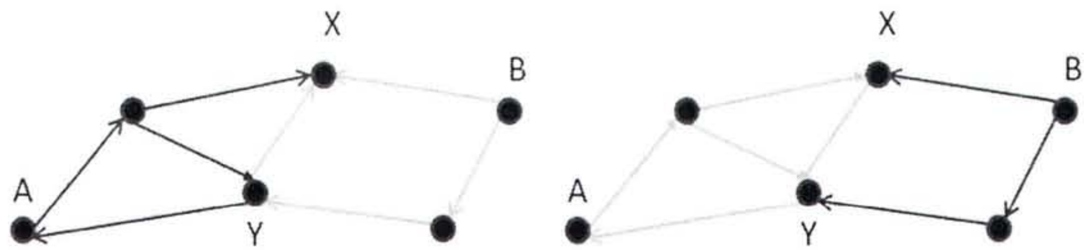


Figure 2.6 In/Out component
(Newman, M.E.J., 2011)

Out component of vertex A is the set of all vertices that can be reached from a directed path beginning from A . In Figure 2.6 (left) the out components of vertex A and vertex B is depicted. Vertices X and Y belong to both. All members of strongly connected components have the same out component. Conversely in component of vertex A is the set of all vertices that can be reached to A by a directed path. As in Figure 2.6 (right) the intersection of in and out components of a vertex is equal to the strongly connected component of it belongs to.

There may be many paths between two vertices. There are two types of independent paths, edge and vertex independent. If a path visit edges between two vertices exactly once then it is edge independent. Similarly if a path visit vertices between two vertices exactly once on its route then it is vertex independent. The number of independent paths between a pair of vertices is said to be the connectivity. In Figure 2.7 edge connectivity is 2 and vertex connectivity is 1.

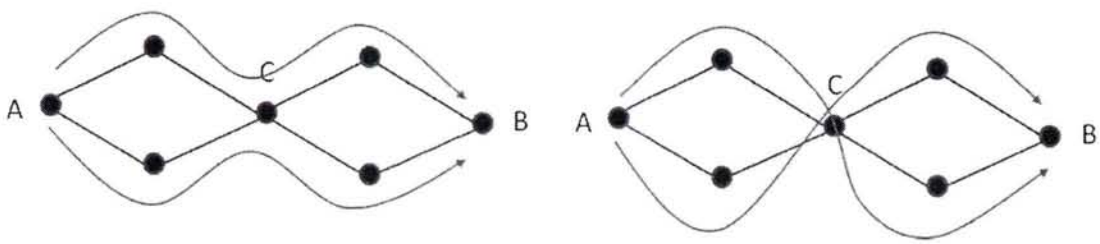


Figure 2.7 Minimum cut sets
(Newman, M.E.J., 2011)

A vertex cut set is a set of vertices whose removal will disconnect a pair of vertices. An edge cut set is the same for removing the edge. In the minimum cut sets are: $\{W,Y\},\{W,Z\},\{X,Y\},\{X,Z\}$. In Figure 2.8, Menger's theorem states that if there is no cut set of size less than n between pair of vertices, then there are at least n independent paths between the same vertices.

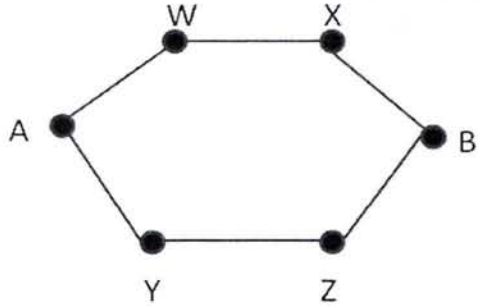


Figure 2.8 Menger's theorem
(Newman, M.E.J., 2011)

Edges can have weights on them representing some edges are stronger. A minimum edge cut set is defined as being a cut set such that the sum of the weights on the edges of the set has the minimum possible value. Maximum flows and minimum cut sets on weighted networks are related with the max-flow/min-cut theorem where the maximum flow between a pair of vertices in a network is equal to the sum of the weights on the edges of the minimum edge cut set that separates both vertices.

Cores, cliques, components, plexes are some of the structures that form social networks. We have mostly used components and cores in our study.

In general, connected parts of a network are called *components*. To better understand the *components* concept it is better to define the terms: *semiwalk*, *walk*, *semipath* and *path*.

A *semiwalk* is the sequence of lines where the end of one line is the starting node of the consecutive line. It is a *walk* when these lines are in a sequence of arcs following the tail and head of each other in a rule.

A *semipath* is a semiwalk where each node should only passed once. Similarly a *path* is a walk where each node should only passed once.

Connectedness now can easily be defined by using the terms described above, where a network is *weakly connected* if each node pairs are connected by semipaths. A network is *strongly connected* if all node pairs are connected by paths.

In undirected networks, components are isolated from each other and there are not any line between each other therefore weakly connected components should be taken into consideration. To analyze the directed networks, strongly connected components can be used for discovering clusters in the network.

If the network consist of one large weak component it is better to split it up into strong components (De Nooy W. et al, 2005).

2.1.3. Cores

Another construct for groups of vertices is *k-core* where *k-core* is a maximal subset of vertices such that each is connected to at least *k* others in the subset. *K-cores* can be used to identify the clusters or cohesive groups in a network by using the degree property of the network. For instance a 2-core contains all nodes that are connected to at least 2 of others (De Nooy W. et al, 2005).

Since two *k-cores* that share one or more vertices will form a larger core, *k-cores* cannot overlap.

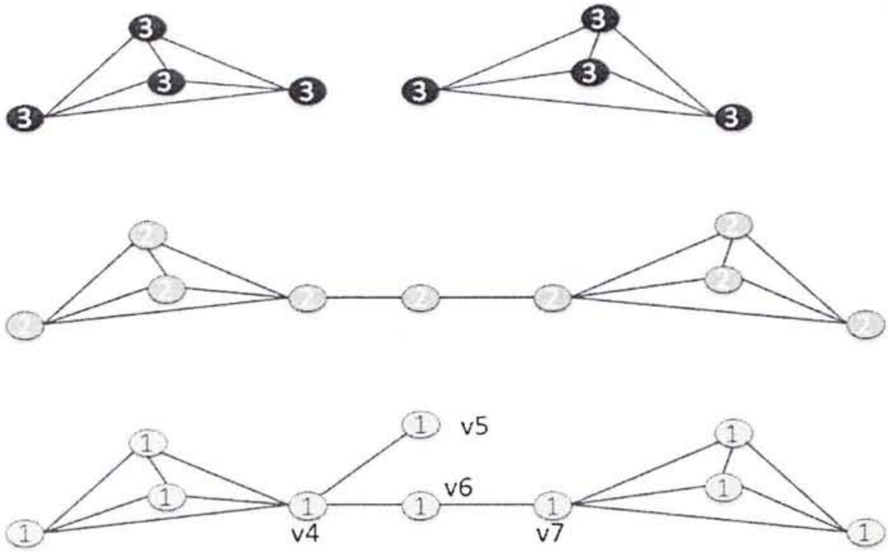


Figure 2.9 Cores
(De Nooy W. et al, 2005)

As shown in the
Figure 2.9 shows the numbers in the figure indicate k for the k-cores. As seen in the 2-core, removing v6 will result in having two clusters as shown in the upper bound of the figure.

In this thesis, it is a preferable strategy to detect clusters by removing the smallest k-cores until the network has dense components.

2.1.4. Cliques

A clique is a maximal complete sub network in an undirected network where every member of the set is connected by an edge to every other. Here maximal means for the clique there is not any vertex in the network that can be added to the k-clique to make it k+1 clique (Newman M.E.J., 2011).

Unlike k-cores cliques may overlap by sharing one or more of the same vertices. An example of a clique of four vertices is shown in the Figure 2.10 .This is a 4-clique where all vertices are connected each other. Overlapping cliques are the densest components of a network and can be accepted as the skeleton of the network (De Nooy W. et al, 2005).

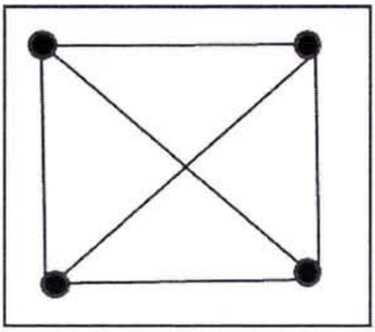


Figure 2.10 Cliques
(Newman M.E.J., 2011)

2.1.5. Plex

In a k-core some members may be unacquainted, even if most members know each other. For this situation a construct named k-plex may help. A k-plex of size p is a maximal subset of p vertices each vertex should be connected to at least $p-k$ of the others. Like cliques K-plexes can overlap. In real life many social groups may form k-plexes. The value k may be selected experimentally. Small values of k may yield meaningful values for small groups (Newman M.E.J., 2011).

2.2. Measures and Metrics

Metrics used in network analysis are listed below.

2.2.1. Degree and Centrality

Degree of a vertex is the number of edges connected to it. Degree of vertex i will be denoted as k_i. The degree of an undirected graph for n vertices is given by:

$k_i = \sum_{j=1}^{n} A_{ij}$ Each edge connecting 2 vertices in an undirected graph will be represented by twice in adjacency matrix. Therefore, there are $2m$ edges in total and $2m = \sum_{j=1}^{n} k_i$

The mean degree for any vertex in an undirected graph is depicted as c where

$c = \frac{1}{n}\sum_{j=1}^{n} k_i = \frac{2m}{n}$. The maximum possible count of edges in a simple graph is depicted with the formula $\binom{n}{2} = (n-1)n\frac{1}{2}$. The connectance or density, σ, is presented as:

$\sigma = \frac{m}{\binom{n}{2}} = \frac{c}{n-1}$. $0 <= \sigma <= 1$. A graph with $\sigma \to 0$ and $n \to \infty$, it is said to be sparse and the fraction of nonzero elements in adjacency matrix also approaches to zero. When σ tends to be constant as $n \to \infty$, a graph is dense.

A regular graph is a graph where all vertices have the same degree. The in-degree of a vertex is the count of all edges directing to it and the out-degree of a vertex is the number of edges directing to other vertices.

Node-based centrality is used for the importance of a node in the network. When node-based centrality score is high for a node, it can be accepted as high influential node. Degree centrality is the number of paths starting from a node. K-path centrality is the number of maximum k paths that start from a node.

2.2.2. Betweenness Centrality

As a median measure Freeman proposed a model for betweenness, how much the node is on the way of shortest paths:

$$c_i^{BET} = \sum_{J=1} \frac{b_{jik}}{b_{jk}}$$ Eq.3 (Freeman L. C., 1979).

b_{jik} is the number of paths passing from j to k and b_{jk} is the number of shortest paths from j to k.

Node betweenness is similar to edge betweenness where the most visited nodes can have critical roles in the networks. If a node is connected to multiple nodes in a network then it is a structural hole. Structural holes are the nodes connecting the discrete regions of a network (Aggarwal C. C., 2011).

2.2.3. Closeness Centrality

The farness of a node is the sum of distances of an actor node x_i to other nodes in a graph. In the meaning, closeness is the inverse of farness. x_i is said to be central if it has short distances to the others. Shortest distance can be used to measure this value where shortest distance from actor i to actor j is denoted as $d(i, j)$ and the closeness centrality formula for undirected graphs is given by:

$$C_c(i) = \frac{n-1}{\sum_{j=1}^{n} d(i, j)}$$ Eq.4 (Liu B., 2007)

The value can be between 0 and 1 where $n-1$ will be the minimum value for the denominator. For the directed graphs the directions of the paths should be taken into consideration (Liu B., 2007).

2.2.4. Katz Centrality

Katz centrality counts the number of walks starting from a node and penalizes longer walks (Katz L., 1953).

$$c_i^{KATZ} = e_i^T (\sum_{J=1}^{\infty} (\beta A)^J 1$$ Eq.5 (Aggarwal C. C., 2011).

e_i Stands for a column vector where i th element is 1 and all other are 0. $0<\beta<1$ is a penalty value.

Katz centrality can be used in bi-directional graphs such as WWW or citation networks for calculation of centrality or influence of nodes (Aggarwal C. C., 2011).

2.2.5. Tie Strength

In (Granovetter M., 1985), the tie strength is explained as the overlap of the neighbors' of the nodes where the increase in the common neighbor number will increase the strength of the tie.

$$S(A,B) = \left| \frac{n_A \cap n_B}{n_A \cup n_B} \right|$$ Eq.6 (Newman, M.E.J., 2011)

n_A is the number of A's neighbors and n_B is the number of B' neighbors
If the overlap for Nodes A-B is small then the tie-strength is low else when there is no overlapping of Nodes A-B then there is local bridge. If the tie A-B is removed and the connection part containing nodes A and B are discrete then this tie is a global bridge.

2.2.6. Triadic Closure

Triadic-closure is a hypothesis about tie-strength. If Nodes A-B and Nodes A-C have strong ties then Nodes B-C is supposed to have strong tie. Triadic- closure is measured by the clustering coefficient of the network $C(p)$.

2.2.7. Clustering Coefficient

Clustering coefficient means the possibility of Node A's randomly selected friends to be friends of each other as well. Let v is the node and k_v is the number of neighbors of v, then $k_v.(k_v - 1)/2$ is the maximum neighbor number of v. $C(v)$ is the fraction of allowed edges And local clustering coefficient for undirected graphs is given by $C(p)$ = 6 / $k_v.(k_v - 1)$ (Watts D.J. and Strogatz S.H., 1998).

Clustering coefficient is the fraction of the paths of size two with the closed ones or in other words is the fraction of transitive triples. Triples here can be described as three vertices uvw with edges uv and vw. There may be 3 triangles for this node sequence. So the clustering coefficient C can be described as:

$$C = \frac{(number of triangles) * 3}{(number of connnected triples)}$$ Eq.7 (Newman, M.E.J., 2011)

When C=1 network has perfect transitivity. When C=0 network can be a tree or square lattice. C is expected to have high values for social networks and dense behavioral networks (Newman, M.E.J., 2011).

2.2.8. Embeddedness

Embeddedness is the value where individuals are enmeshed in a social network. In other words it is the likelihood of a triplet being closed by a tie so that it forms a triangle. Embeddedness is another way of describing tie strength. When two nodes are connected with an embedded edge, they can trust each other, because there are common people to be informed about each other. If there is not any embedded edge they have no common friends (Granovetter M., 1985).

2.2.9. Transitivity

Transitivity is a term defined in mathematics and is related to the 'friend of my friend can be my friend' concept. For equality if a=b and b=c then a=c. In network means if node u is connected to node v and node v is connected to node w then it is more likely for u to be connected to node w, according to a randomly chosen node.

Perfect transitivity can occur when all nodes in a network are connected to each other. This may not be very useful for network discovery. But partial transitivity may work where u knows v and v knows w they form a path uvw. When u and w are connected they form a *closed triad* (Newman, M.E.J., 2011).

2.2.10. Homophily

Homophily is the phenomenon that refers to the selection of the friends of a person according to their similar characteristics such as gender, ethnicity, nationality and appearance (Ruef et al., 2003).

Three main elements that form Homophily are:

- Social Influence : Behavioral change of an actor that is influenced by another actor in the social group
- Selection: In a social group, members with similar characteristics tend to group together.
- Confounding Variables: Other variables for members who tend to behave similar.

Selection can be used for recommendation systems while social influence can be used for viral marketing (Aggarwal C. C., 2011).

3. SOCIAL NETWORK ANALYSIS

3.1. Social Networks

In general, a social network can be defined as a network where actors are nodes and edges are the relationships such as friendship, common interest, relationship of beliefs etc.

'Social' and 'Network' words are combined to express the Social Network concept. To better understand Social Network concept 'Social' behaviors and 'Network' structures can be investigated diversely.

Social Networks are emerging as a new research area gathering many disciplines such as sociology, computer science and mathematics. In today's world Web 2.0 applications such as Facebook and LinkedIn, micro blogging applications like Twitter are good examples of social network structures. Also Social Networks can be identified in Mobile or Landline telephone networks, social clubs and customer chains.

Real world problems can be represented in different relationship model networks where entity-relationship structures can be observed. These networks can be engineering, linguistic, ecological, and biological vice versa. 'Network Science' is to observe and expose the common properties of the social network where those network types share common behaviors (Aggarwal C. C., 2011).

Content generated by the Web 2.0 applications like Facebook, Twitter and Flickr can be used for many types of applications. One example is customer feedback where the customers have the chance to be informed each other for reviews, opinion sharing etc.

Community discovery can help to understand the social structure of the network, help in answering the questions such as 'How the network evolves?' In networks, there are nodes with greater ties with each other than to the rest of the network forming a network part called 'communities'. These communities can be discovered by community discovery methods that can be used in viral marketing, churn prediction and ratings predictions (Aggarwal C. C., 2011). Community discovery algorithms can be used to define communities and they can be different in accordance to their approach to the problem, performance, user intervention, balanced division.

In the recent years social network approach has been increasingly applied in computer science disciplines. With the advance in web technologies, there is (Kumar et al., 2003) greater amount of interaction by people interacting on the Internet. Social Networks come in a multi-disciplinary approach to solve problems in this environment. The Internet gives us new questions about the nature of social networks and provides new perspectives for social network analysis.

A number of studies have analyzed patterns of linking on the World Wide Web. In (Adamic L. A., 1999) linking patterns of WWW have been analyzed and WWW is accepted as a small world network (Gibson et al., 1998) as proposed, a method to detect hubs and authorities in WWW.

Internet Relay Chat is a system that allows people to collaborate and chat from any location in the world. Mutton (Mutton P., 2004) has proposed a model that uses an IRC bot that monitors the channels and creates a mathematical model of the social network by using heuristic methods. Thus, the bot can produce a visualization of the social network. Those kind of visualizations, exposure the structure of the social network, by connectivity, clustering and communications between users in the IRC. Animated output in the study shows the social network in a time evolving fashion.

These days, SNA methods have begun to be used for Weblogs where people can have online social communication. (Kumar et al., 2003) observed and modeled the connectivity within blog groups and he concluded that not in scale also in connectedness means these kinds of networks are growing.

Marlow (Marlow C, 2006) uses social network analysis to quantitatively analyze and visualize link patterns of authoritative blog authors, and compare them with leadership and authority metrics. The study was implemented by checking the links between and referrals each other. As a result some blog lists were central and other blog groups were in dense structure.

Mobile call graphs are scale free graphs in similarity with power law distributions. In a research conducted by (Nanavati et al., 2007) call graphs are defined in a model named 'Treasure Hunt' model in purpose of observing and defining the certain parameters and topology of this kind of graphs. This model is based on the idea of analyzing the edges of call graphs which may follow a pattern rather than analyzing the nodes. In this kind of analysis, cliques (closed exclusive group sharing common interests, political view, behavior etc.) are discovered and patterns are analyzed.

In (Richter et al., 2010), a prediction model is proposed named 'group-first churn prediction' in the idea of analyzing social influence in customer groups. Their hypothesis claims that in spite of the fact that there are closely grouped structures in mobile networks, positive and negative feedback is rapidly propagated through these small groups and these groups tend to be a subscriber of the same mobile carrier. The implementation is started by analyzing mobile customers using second order social metrics in closely grouped structures and after all interactions within each group is analyzed to find out social leader of the corresponding group and statistical models are used to assign a risk score for each group.

Selected KPIs for each group are developed by using machine learning techniques to fit group churn. Finally personal churn scores are assigned for each member depending on his group score.

In the year 1967, Stanley Milgram has executed a study to prove the small – world problem. Small–world problem can be described as: How many intermediate acquaintances are required to reach from a random chosen person A to random chosen person B?

The experiment is funded by Harvard University. The methodology was to select a group of random people living in the different places of the United States and request them to forward a message to the same target person. A folder has forwarded to each receiving person including the target person's address information and a bucket of rosters for sent confirmation. There were some rules to take care of:

- Messages should be sent to the next person who they know in the first name basis.
- Message should be forwarded to the most likely person to be able to find the target.
- Each person should return a roster to the research center after he forwards the message.

The result of the study was:

- The median value of the chains was 5.
- Some of the chains were completed and some were not.
- Participants were more likely to send the message to someone of the same sex.
- Most intermediate senders were friends not relatives. This can change according to the social structure of the networks.
- Not all the people in the ring have the same social influence value. The target person received all messages from 3 different people (last people in the chain) (Milgram S, 1967).

3.2. Community Discovery & Graph Partitioning Algorithms

Some of the community discovery algorithms use graph partitioning methods. Graph partitioning is the problem of dividing a network into fixed size non-overlapping pieces to minimize the interconnecting edges. In other words, a partition in a network is a construct where each vertex belongs to one class or cluster. By graph partitioning it is easier to reduce a network's size and complexity (De Nooy W. et al, 2005). Community detection is similar but different concept from graph partitioning where groups and size of the groups is not fixed as in graph partitioning. Detection is done more naturally and the parameters are set by the network itself.

Different algorithms are used for graph partitioning and graph clustering.
We give a list of algorithms in section 3.2.1. In section 3.2.2 we reviewed some of the important algorithms in detail. The algorithms reviewed include Kernighan Lin, Spectral Partitioning, Newman Edge Betweenness algorithm, MCL algorithm, Hierarchical Clustering algorithm and K-core Community Discovery method.

3.2.1. A list of Community Discovery Algorithms

A list of most popular community discovery algorithms is listed below:
(Aggarwal C. C., 2011).

Algorithm Type	Description	Paper name	Reference
Edge Betweenness Algorithm		Community structure in social and biological networks	Girvan, M., and M. E. J. Newman, 2002, Proc. Natl. Acad. Sci. USA 99(12), 7821.
Kernighan-Lin Algorithm	The authors were motivated by the problem of partitioning electronic circuits onto boards: the nodes contained in	An efficient heuristic procedure for partitioning graphs An algorithm for	Kernighan, B. W., and S. Lin, 1970, Bell System Tech. J. 49, 291. Suaris, P. R., and G. Kedem, 1988, IEEE

	different boards need to be linked to each other with the least number of connections.	quadrisection and its application to standard cell placement	Trans. Circuits Syst. 35, 294.
Spectral Bisection algorithm	It is based on the properties of the spectrum of the Laplacian matrix	An algorithm for partitioning the nodes of a graph	Barnes, E. R., 1982, SIAM J. Alg. Discr. Meth. 3, 541.
Max-flow Min-cut Algorithm	This theorem has been used to determine minimal cuts from maximal ows in clustering algorithms. In the Flake's paper it used maximum flows to identify communities in the graph of the World Wide Web.	A new approach to the maximum-flow problem Self-organization and identification of web communities	Goldberg, A. V., and R. E. Tarjan, 1988, Journal of the ACM 35, 921. Flake, G. W., S. Lawrence, C. Lee Giles, and F. M. Coetzee, 2002, IEEE Computer 35, 66.
Level – Structure Partitioning	This algorithm computes vertex seperators that was provided in Sparspak, a library of routines for solving sparse systems of equations by direct methods.	Graph partitioning algorithms with applications to scientific computing	Pothen, A., 1997, Graph Partitioning Algorithms with Applications to Scientific Computing, Technical Report, Norfolk,VA, USA.
Inertial Algorithm	The Inertial Algorithm employs the geometrical coordinates of the vertivces of a graph embedded in two or three dimensions to compute a partition.	Graph partitioning algorithms with applications to scientific computing	Pothen, A., 1997, Graph Partitioning Algorithms with Applications to Scientific Computing, Technical Report, Norfolk,VA, USA.
Spectral Clustering Algorithm	Spectral clustering consists of a transformation of the initial set of objects into a set of points in space,	Spectral K-Way Ratio-Cut Partitioning and Clustering	Chan, P. K., M. D. F. Schlag, and J. Y. Zien, 1993, in Proceedings of the 30th International Conference on Design

	whose coordinates are elements of eigenvectors		Automation (ACM Press, New York, USA), pp. 749-754.
		A New Approach to Effective Circuit Clustering	Hagen, L., and A. B. Kahng, 1992, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 11(9), 1074.
		Lower bounds for the partitioning of graphs	Donath, W., and A. Ho_man, 1973, IBM Journal of Research and Development 17(5), 420.
		A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory	Fiedler, M., 1973, Czech. Math. J. 23(98), 298.
		Normalized Cuts and Image Segmentation	Shi, J., and J. Malik, 1997, in CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97) (IEEE Computer Society, Washington, DC, USA), p. 731.
		On Spectral Clustering: Analysis and an algorithm	Ng, A. Y., M. I. Jordan, and Y. Weiss, 2001, in Advances in Neural Information Processing Systems, edited by T. G. Dietterich, S. Becker, and Z. Ghahramani (MIT Press, Cambridge, USA), volume 14.

Hierarchical Clustering Algorithm	Social networks, for instance, often have a hierarchical structure. Hierarchical clustering is very common in social Network analysis, biology, engineering, marketing, etc. The starting point of any hierarchical clustering method is the denition of a similarity measure between vertices. After a measure is chosen, one computes the similarity for each pair of vertices, no matter if they are connected or not.	The Elements of. Statistical Learning	Hastie, T., R. Tibshirani, and J. H. Friedman, 2001, The Elements of Statistical Learning (Springer, Berlin, Germany), ISBN 0387952845.
K-means Clustering	The distance is a measure of dissimilarity between vertices. The goal is to separate the points in k clusters such to maximize/minimize a given 20 cost function based on distances between points and/or from points to centroids	Comparative study of discretization methods of microarray data for inferring transcriptional regulatory networks Least squares quantization in PCM. A direct approach to	MacQueen, J. B., 1967, in Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability, edited by L. M. L. Cam and J. Neyman (University of California Press, Berkeley, USA), volume 1, pp. 281-297. Lloyd, S., 1982, IEEE Trans. Inf. Theory 28(2), 129. Hlaoui, A., and S. Wang, 2004, in Neural Networks and Computational

		graph clustering. Neural Networks and Computational Intelligence Graph clustering with network structure indices. Graph-Theoretic Techniques for Web Content Mining	Intelligence, pp. 158-163. Rattigan, M. J., M. Maier, and D. Jensen, 2007, in ICML '07: Proceedings of the 24th international conference on Machine learning (ACM, New York, NY, USA), pp. 783-790. A. Schenker, H. Bunke, M. Last, A. Kandel, "Graph-Theoretic Techniques for Web Content Mining", World Scientific, Series in Machine Perception and Artificial Intelligence, Vol. 62, 2005.
Fuzzy k-means Clustering	a point may belong to two or more clusters at the same time and is widely used in pattern recognition.	Pattern recognition with fuzzy objective function algorithms A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters	Bezdek, J. C., 1981, Pattern Recognition with Fuzzy Objective Function Algorithms (Kluwer Academic Publishers, Norwell,USA). Dunn, J. C., 1974, J. Cybernetics 3, 32.
Girvan and Newman Algorithm	Girvan and Newman focused on the concept of betweenness, which	Community structure in social and biological	Girvan, M., and M. E. J. Newman, 2002, Proc. Natl. Acad. Sci. USA 99(12), 7821.

	is a variable expressing the frequency of the participation of edges to a process.	networks Finding and evaluating community structure in networks A method for finding communities of related genes An Introduction to Community Detection in Multi-layered Social Network Graph Clustering with Network Sructure Indices Betweenness-based decomposition methods for social and biological networks	Newman, M. E. J., and M. Girvan, 2004, Phys. Rev. E 69(2), 026113. Wilkinson, D. M., and B. A. Huberman, 2004, Proc. Natl. Acad. Sci. U.S.A. 101, 5241. Tyler, J. R., D. M. Wilkinson, and B. A. Huberman, 2003, in Communities and technologies (Kluwer, B.V., Deventer, The Netherlands), pp. 81-96. Rattigan, M. J., M. Maier, and D. Jensen, 2007, in ICML '07: Proceedings of the 24th international conference on Machine learning (ACM, New York, NY, USA), pp. 783-790. Pinney, J. W., and D. R. Westhead, 2006, in Interdisci- plinary Statistics and Bioinformatics (Leeds University Press, Leeds, UK), pp. 87-90.
Clique Percolation Method (CPM)	It is based on the concept that the internal edges of a community are likely to form	Uncovering the overlapping community structure of complex	Palla, G., I. Der_enyi, I. Farkas, and T. Vicsek, 2005, Nature 435, 814.

	cliques due to their high density.	networks in nature and society Weighted network modules Biclique communities Overlapping community detection in bipartite networks	Farkas, I., D.Abel, G. Palla, and T. Vicsek, 2007, New J. Phys. 9, 180. Lehmann, S., M. Schwartz, and L. K. Hansen, 2008, Phys. Rev. E 78(1), 016108. Du, N., B. Wang, B. Wu, and Y. Wang, 2008, in IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (IEEE Computer Society, Los Alamitos, CA, USA), pp. 176-179.
Markov Clustering Algorithm		miRBase: microRNA sequences, targets and gene nomenclature	AJ Enright, S Van Dongen- Nucleic acids research, 2002 - Oxford Univ Press

3.2.2. Some of the commonly used Community Discovery Algorithms

Some of the most used algorithms in community discovery previously listed above are described below. In this thesis, we have used K-core Community Discovery Method.

3.2.2.1. Kernighan Lin (KL) Algorithm

In (Kernighan B. W. and Lin S., 1970), the problem of partitioning a graph by considering the edge weights and to minimize the cost value in each cut has been researched.

Kernighan Lin (KL) is a greedy algorithm that minimizes the edge cut while keeping cluster sizes balanced. The aim is to partition the graph in two parts by minimizing the cut edges. The algorithm starts with dividing the graph into two parts. This can be achieved manually or randomly. Process goes on by swapping each node pair that reduces the cut size by the largest amount or increases it by the smallest amount. Any swapped node pair should not swap again in each round. This process goes on until no pairs left to be swapped. At last, all states of the network observed and the state in which least number of edge cut will happen, will show the best partitions for division.

Letting (A, B) be an initial partition where $a \in A$ and $b \in B$. The pseudocode for KL algorithm is shown in

```
Compute  T = cost(A,B) for initial A, B
     Repeat
          Compute costs D(n) for all n in N
          Unmark all nodes in N
          While there are unmarked nodes
               Find an unmarked pair (a,b) maximizing gain(a,b)
              Mark a and b (but do not swap them)
              Update D(n) for all unmarked n,
                    as though a and b had been swapped
           Endwhile

          Pick m maximizing Gain = Sk=1 to m  gain(k)
          If Gain > 0 then   … it is worth swapping
               Update newA = A - { a1,…,am } U { b1,…,bm }
               Update newB = B - { b1,…,bm } U { a1,…,am }
               Update T = T - Gain
          endif
     Until Gain <= 0
```

Figure 3.1 Pseudo code for Kerninghan Lin Algorithm
http://parlab.eecs.berkeley.edu/wiki/_media/patterns/graph_partitioning.pdf

Performance is a problem for KL algorithm. Number of swaps for one round is $\frac{1}{2}n \times \frac{1}{2}n = \frac{1}{4}n^2 = O(n^2)$ while there are $O(n)$ swaps in the worst case. Total time for one round of the KL algorithm is $O(n \times n^2 \times \frac{m}{n}) = O(mn^2)$ which is $O(n^3)$ on a sparse network and $O(n^4)$ on a dense network where m is the total number of edges.

KL algorithm has $O(n^3)$ performance and can be easily used for graphs of a few hundreds of thousands of vertices (Newman, M.E.J., 2011).

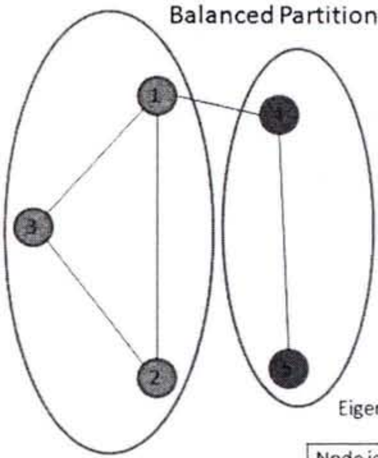
3.2.2.2. Spectral Partitioning Algorithms

Spectral Partitioning Algorithms are another type of divisive algorithms. They can be easily solved by using linear algebra. By using eigenvectors, normalized and unnormalized cuts can be implemented on Laplacian matrix L. $x_1, x_2 .... x_n$ are the data points of the similarity graph of $G = (V, E)$ and $s_{i,j}$>=0 is the similarity. If $s_{i,j}$ defining data points x_i and x_j, is positive or greater than a threshold value then x_i and x_j are connected. W is the adjacency matrix of $G = (V, E)$ where G is an undirected and weighted graph.

$L = D - W$ where D is the diagonal matrix of the nodes of the graph $G = (V, E)$.

When the unnormalized Laplacian is computed, first k eigenvectors are computed. And at last step clusters $C_1, C_2 .... C_n$ will be composed by using the k-means algorithm. For normalized spectral clustering first k generalized eigenvectors should be used (Von Luxburg U., 2007).

As an example, in Figure 3.2 the second smallest eigenvalue (λ) in red marked area gives a better and balanced cut result where (1,2,3) and (4,5) are two communities.



L=D-A

Node id	1	2	3	4	5
1	3	-1	-1	-1	0
2	-1	2	-1	0	0
3	-1	-1	2	0	0
4	-1	0	0	2	-1
5	0	0	0	-1	1

Eigen value decomposition of L: (V)

Node id	1	2	3	4	5
1	-0.44721	0.201774	-0.317515	0	0.8114622
2	-0.44721	0.41931	0.242173	-0.707106	-0.255974
3	-0.44721	0.41931	0.24217	0.7071067	-0.2559747
4	-0.44721	-0.3379	-0.7030	0	-0.4375313
5	-0.447958	-0.70246	0.5362	0	0.13801875
	E=[0,	0.5188,	2.3111,	3.0000,	4.1701]

Figure 3.2 Spectral Partitioning Example
(http://en.wikipedia.org/wiki/Graph_partition)

Spectral clustering has big computational complexity and the main idea is to transform the original graph into a low dimensional format.

3.2.2.3. Newman's Edge Betweenness Algorithm

To find the communities in a network, Newman proposed a divisive method using *betweenness* as a measure. Betweenness is a measure which favors edges that lie between communities and unfavors the ones inside the communities. Three of various types of betweenness measures are shortest-path betweenness, random-walk betweenness current flow betweenness. Shortest-path betweenness is the sum of all shortest geodesic paths between all pairs of vertices. Shortest-path betweenness can be thought of as the signals travelling through a network where all vertices can send signals at the same time. However signals may not follow geodesic paths and they can perform random walks. This can be identified as random-walk betweenness where it can be calculated as the net number of times that a random walk between a particular pair of vertices will pass down a particular edge and sum over all vertex pairs. Current flow betweenness can be calculated using

Kirchhoff's laws in the imagination of the network as a circuit where edges are resistance and nodes are sinks.

Algorithm performs by calculating edge betweenness for each edge in the network and removing edges in the decreasing order of betweenness to produce a dendogram. When an edge in the network is removed, the betweenness values for the remaining edges are recalculated. As an example in Figure 3.3, the thickness of the edge line is higher when the betweenness value is high. The thickest line between nodes is on all paths between nodes in the two different communities so it has a high edge betweenness.

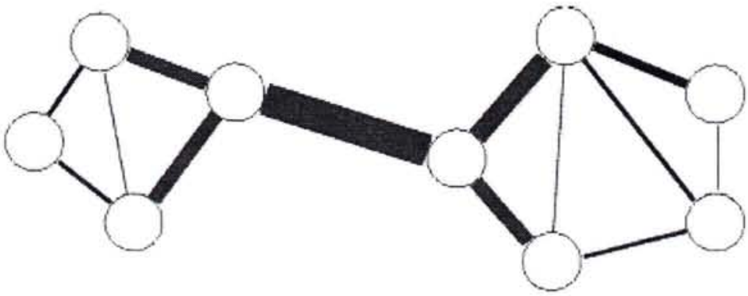


Figure 3.3 An example of betweenness
http://discopal.ispras.ru/SocialGraphs/Community_Detection

The steps of the community structure finding algorithm:

1. Calculation of betweenness scores for all edges in the network.
2. Define the edge with the highest score and delete it from the network.
3. Recalculation betweenness for all remaining edges.
4. Repeat from step 2.

In Figure 3.4 there is an example of a community discovery analysis executed by Newman's edge betweenness algorithm:

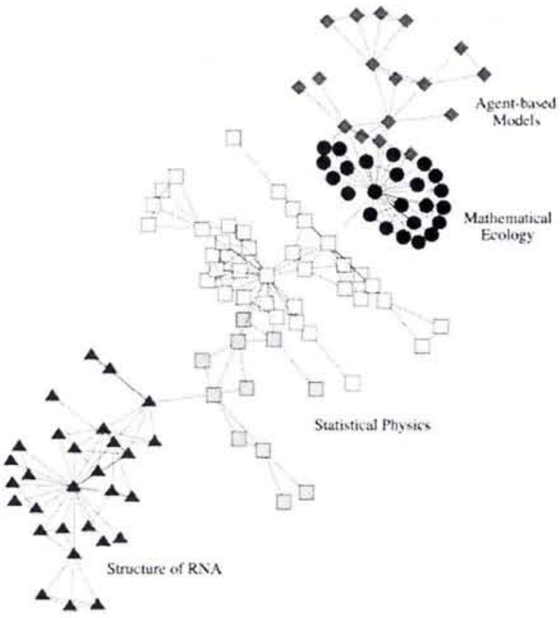


Figure 3.4 The largest component of the Santa Fe Institute collaboration network, with the primary divisions detected by algorithm indicated by different vertex shapes.
(Girvan M. and Newman M. E. J, 2002)

Calculation of the edge betweenness measure based on geodesic paths for all edges will take $O(mn^2)$ or $O(n^3)$ time on a sparse graph calculating the shortest path between a particular pair of vertices can be done using breadth-first search in time $O(m)$ and there are $O(n^2)$ vertex pairs (Newman M.E.J. and Girvan M., 2004).

3.2.2.4. Markov Clustering Algorithm (MCL)

Markov Clustering Algorithm was invented by Stijn van Dongen, scalable unsupervised cluster algorithm for graphs, executes in two steps: Expand and Inflate. By doing Random walks on a graph it will be possible where flow is collected and starting from a node the traveler will more likely tend to stay in the strongly connected clusters. Random walks are calculated by using "Markov Chains". These values are collected in a stochastic matrix.

To apply MCL, expansion and inflate methods can be applied to the graph many times and a smaller graph is obtained. The main idea here is to apply a matching strategy where none of the two edges are belonging to the same vertices; these vertices are collapsed. Randomized strategies can be used for expansion.

Expansion step is for spreading the flow to the other new vertices and helps in spreading the flow to reachable vertices in multiple steps. Within cluster flow will increase in the idea that there are many paths for the vertices in the same cluster. Expansion and Inflation matrices both map the column space stochastic matrices on to themselves. Expansion and Inflation are executed iteratively.

Expansion can be described as below:

$Expand: M_{\exp} = Expand(M) = M * M$ Eq.8 (Satuluri V. and Parthasarathy S., 2009).

Inflation is applied for inhomogenition of the Deflated matrix where the flow is stronger, it will be strengthened and where the flow is weaker it will be weakened.
Inflation can be described as below:

$Inflate: M_{\inf}(i,j) = M(i,j)^r \dfrac{M(i,j)^r}{\sum_{k=1}^{n} M(k,j)^r}$ Eq.9 (Satuluri V. and Parthasarathy S., 2009).

By default inflation parameter r=2, $M_{\inf}$ corresponds to raising each entry value in M matrix to the power r and then normalizing the matrix column values to 1.
Pruning is applied to amend the computation time by removing very small values in each column and recalculating to provide all column values to be equal to 1. Prune threshold values will be smaller than the maximum and average column heuristic values.
Pseudo code for MCL is shown in Figure 3.5

Algorithm 1 MCL

$A := A + I$ // Add self-loops to the graph
$M := AD^{-1}$ // Initialize M as the canonical transition matrix

repeat
  $M := M_{exp} :=$ Expand(M)
  $M := M_{inf} :=$ Inflate(M, r)
  $M :=$ Prune(M)
until M converges

Interpret M as a clustering

Figure 3.5 Pseudo code for MCL Algorithm
(Newman, M.E.J., 2011)

MCL has lack of scalability problem and MCL is very time consuming because of the multiplication processes during Expansion stage. Expansion can be done in $O(n^2)$ time.

As another limitation; MCL can lead to unbalanced partitions: many small partitions with few vertices or producing a very big one, or both situations can happen at the same time (Satuluri V. and Parthasarathy S., 2009).

3.2.2.5. Hierachical Clustering Algorithm

Hierarchical clustering is one of the oldest community detection methods that produce hierarchical decomposition. Hierarchical clustering is an agglomerative algorithm starts with individual vertices and joins them together in groups.

The main idea is to define a similarity or connection strength metric for vertices and join together the most similar vertices to compose groups.

As metrics, cosine similarity, correlation coefficients between rows of the adjacency matrix or Euclidian distance. Generally the selection of the measure is determined by experience or experiment.

We need to combine vertex similarities to create similarity scores for groups. There are three common ways to achieve this: single-, complete- and average linkage clustering. For example when we consider two groups A and B, n_1 and n_2 vertices respectively in the single linkage clustering method the similarity between the groups A and B will be the most similar of these $n_1 n_2$ pairs of vertices. On the other side complete linkage clustering method defines the similarity value as the least similar pair of vertices. In between these two methods average linkage clustering method is defined to be the mean similarity of all pairs of vertices.

The general algorithm for hierarchical clustering method is:

1. Choose a similarity measure and evaluate it for all vertex pairs.
2. Assign each vertex to a group of its own, consisting of just that one vertex. The initial similarities of the groups are simply the similarities of the vertices.
3. Find the pair of groups with the highest similarity and join them together into a single group.
4. Calculate the similarity between the new composite group and all others using one of the three methods (single-,complete-, or average linkage clustering)
5. Repeat from step 3 until all vertices have been joined into a single group.

As before the groups A and B to be joined they have n_A and n_B vertices where the similarities of A and C and B and C were previously σ_{AC} and σ_{BC} then the composite group's similarity is given by the weighted average:

DOĞUŞ ÜNİVERSİTESİ

$$\sigma_{AB,C} = \frac{n_A\sigma_{AB} + n_B\sigma_{BC}}{n_A + n_B}$$ Eq.10 (Newman, M.E.J., 2011)

As an example: a hierarchical clustering of distances in kilometers between some Italian cities. The method used here is single-linkage. Input distance matrix (L = 0 for all the clusters):

	BA	FI	MI	NA	RM	TO
BA	0	662	877	255	412	996
FI	662	0	295	468	268	400
MI	877	295	0	754	564	138
NA	255	468	754	0	219	869
RM	412	268	564	219	0	669
TO	996	400	138	869	669	0

The nearest pair of cities is MI and TO, at distance 138. These are merged into a single cluster called "MI/TO". The level of the new cluster is L(MI/TO) = 138 and the new sequence number is m=1. Then the distance from this new compound object to all other objects. In single link clustering the rule is that the distance from the compound object to another object is equal to the shortest distance from any member of the cluster to the outside object. So the distance from "MI/TO" to RM is chosen to be 564, which is the distance from MI to RM, and so on.

After merging MI with TO we obtain the following matrix:

	BA	FI	MI/TO	NA	RM
BA	0	662	877	255	412
FI	662	0	295	468	268
MI/TO	877	295	0	754	564
NA	255	468	754	0	219
RM	412	268	564	219	0

min d(i,j) = d(NA,RM) = 219 => merge NA and RM into a new cluster called NA/RM
L(NA/RM) = 219
m = 2

	BA	FI	MI/TO	NA/RM
BA	0	662	877	255
FI	662	0	295	268
MI/TO	877	295	0	564
NA/RM	255	268	564	0

min d(i,j) = d(BA,NA/RM) = 255 => merge BA and NA/RM into a new cluster called BA/NA/RM

L(BA/NA/RM) = 255

m = 3

	BA/NA/RM	FI	MI/TO
BA/NA/RM	0	268	564
FI	268	0	295
MI/TO	564	295	0

min d(i,j) = d(BA/NA/RM,FI) = 268 => merge BA/NA/RM and FI into a new cluster called BA/FI/NA/RM

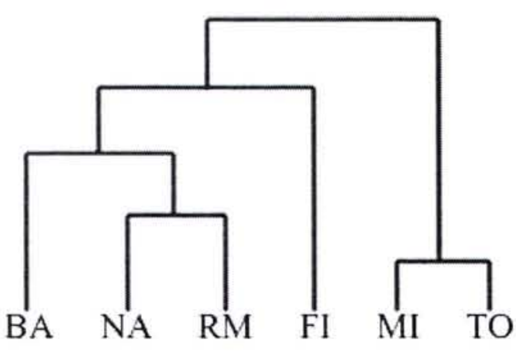
L(BA/FI/NA/RM) = 268

m = 4

	BA/FI/NA/RM	**MI/TO**
BA/FI/NA/RM	0	295
MI/TO	295	0

Finally, we merge the last two clusters at level 295.

The process is summarized by the following hierarchical tree:



http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/hierarchical.html

The total running time of the algorithm is $O(n^3)$ in the naïve implementation or $O(n^2 \log n)$ if we use heap (Newman, M.E.J., 2011).

3.2.2.6. K-core Community Discovery Method

It is possible to discover cohesive groups, in other words: communities by applying k-cores described in section 2.1.3. As mentioned before, k indicates the minimum degree of each vertex within the core. For instance a 2-core contains two degree vertices connected to the other vertices in the core. A k-core may help discovering the communities by identifying relatively the dense subnetworks. In this thesis this methodology is used.

In the sample network in
Figure 3.6, 0,1,2 and 3-cores can be seen.

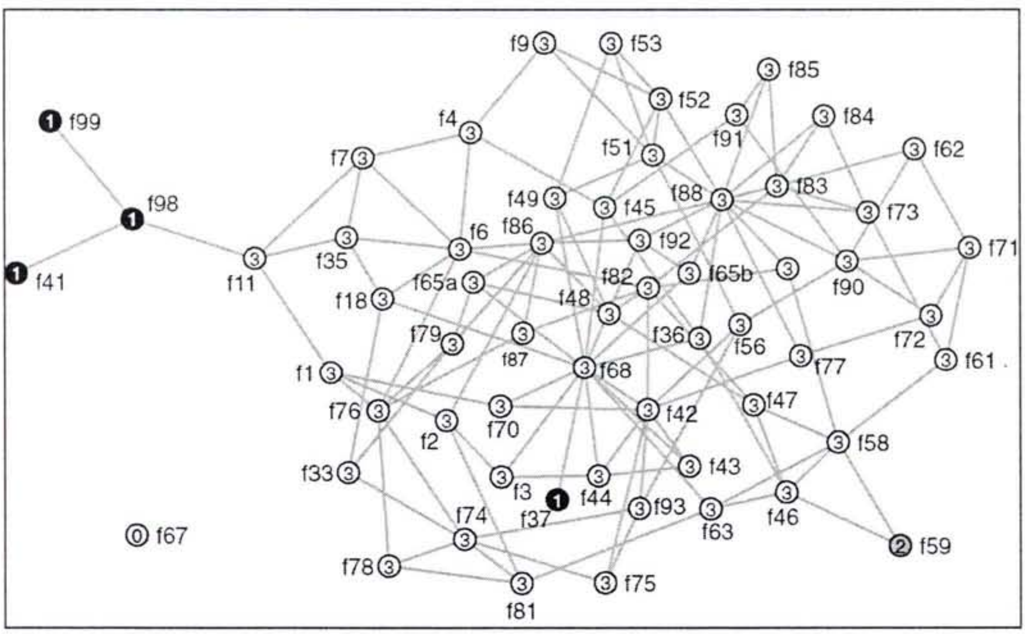


Figure 3.6 A sample network
(De Nooy W. et al, 2005).

In Figure 3.7, vertex v6 can be removed to obtain a more dense network which includes 3-cliques.

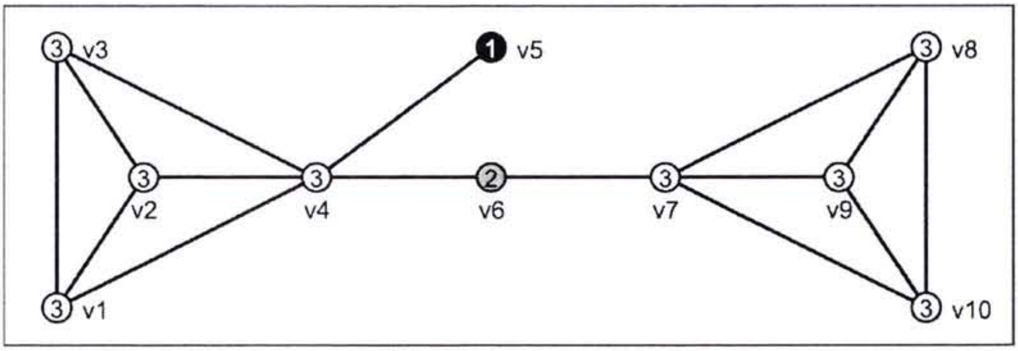


Figure 3.7 A sample graph of 3-cores
(De Nooy W. et al, 2005)

This is a method that can be used to detect cohesive subgroups or communities; simply remove the lowest k-cores from the network until the network breaks up into relatively dense components, preferabaly cliques. As a result, each component can be thought as a cohesive subgroup or community in social science. In large networks, this is an effective way of detecting communities. Iteratively it is possible to increase the level of k-cores and refining the community graph by appyling stong or weak component transformation as defined in the Figure 3.8

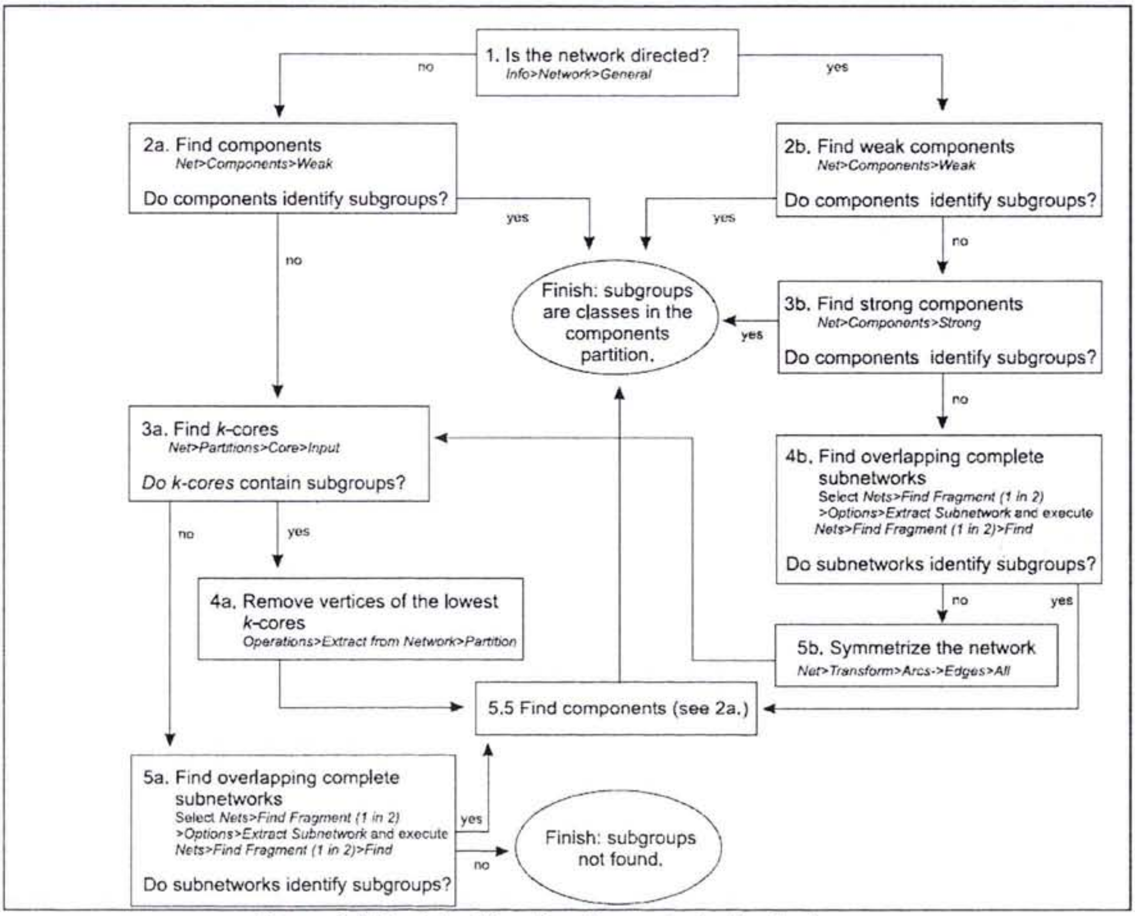


Figure 3.8 Decision Tree for the analysis of cohesive groups
(De Nooy et al., 2005).

3.2.2.7. Main Path Analysis Method

In principle, articles can cite only previous published articles. Because of this the citation networks should be acyclic. Sometimes there can be exceptions for the articles that have been written at the same time citing each other. These can form loops.

Nowadays, citations are used to describe the importance of papers, authors. Citation analysis can be used to reveal the evolution of research traditions. Citation analysis can find out communities formed by researchers of a particular area.

A special technique named main path analysis was proposed by N. Hummon for citation analysis. The idea is that most important citations form one or more main paths of a research tradition. Main path analysis achieves this by calculating the traversal weight of a citation. The procedure counts all the paths from source nodes to sink nodes. After that it counts the paths that include a particular weight. And it divides particular weight to the total and thus finds out the traversal weight of a citation.

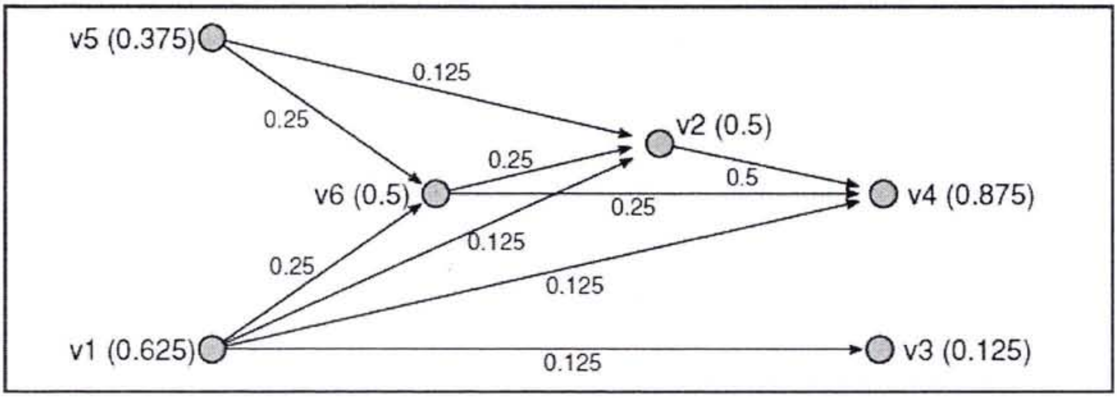


Figure 3.9 Traversal weights in a citation network
(De Nooy et al., 2005)

As an example in Figure 3.9, there are two sources (v1, v5) and two sinks (v4, v3). A path connects v1 and v3. The total number of paths is 8. So the traversal weight for this connection is 1/8 (0.125). All traversal values in Figure 3.9 care calculated in this way. And the next step is to extract the main paths which identify the main flow of a literature. In a citation network, the main path can be defined as the path that has the highest traversal weights from a source to sink node. The main path here starts with v1 and v5 because they have the same values: 0.25. v6 is the next vertex on the main path. After v6 the main paths direct to v4 or via v2 to v4. One main path leading to the same sink can be accepted as a research tradition.

In Pajek, to define the main path, at first all the loops must be removed. And after that the normalization method should be selected according to the weight values. SPC (Search Path Count) command can now be applied to find the main path. 'Line Values' command will list the ranges for the edge weights. A cut-off value between 0 and 1 (trivial) can be selected and the lines smaller than this cut-off values should be removed. (De Nooy et al., 2005).

3.3. Tools for Social Network Analysis

General tools used in Network analysis are listed below. We have used Pajek for analysis and Gephi for metrics calculation.

3.3.1. Tools in General

SNA tools can be grouped in two, one category of tools is specialized in only visualizing the graphs and the other category can have both analysis and visualization capabilities. Most of the tools used in social network analysis are listed in Table 3.1 .

Name	Availability	Platform	Description
Pajek	Free	W	Interactive social network analysis and visualization
Gephi	Free	W	Interactive network analysis and visualization
Net Workbench	Free	WML	Interactive network analysis and visualization
Netminer	Commercial	W	Interactive social network analysis and visualization
InFlow	Commercial	W	Interactive social network analysis and visualization
UCINET	Commercial	W	Interactive social network analysis
yEd	Free	WML	Interactive Visualization
Graphviz	Free	L	Visualization
NetworkX	Free	WML	Interactive network analysis and Python library
JUNG	Free	WML	Java library for network analysis and visualization

İgraph	Free	WML	C/R/Python libraries for network analysis
GTL	Free	WML	C++ library for network analysis
LEDA/AGD	Commercial	WL	C++ library for network analysis

Table 3.1 SNA Tools
(Newman, M.E.J., 2011)

In this thesis, Pajek and Gephi tools are being used and the detailed information about these tools can be found in the next section.

3.3.2. Pajek

In this thesis, as a tool, Pajek is used for clustering the social networks. Pajek is a Windows program for analysis and visualization of large networks having some thousands or even millions of vertices. The recent version of Pajek can be downloaded from:
http://vlado.fmf.uni-lj.si/pub/networks/pajek/

Pajek is developed with Pascal in 1996. Pajek provides tools for analysis and visualization of such networks: collaboration networks, organic molecule in chemistry, protein-receptor interaction networks, genealogies, Internet networks, citation networks, diffusion (AIDS, news, innovations) networks, data-mining (2-mode networks), etc.
http://vlado.fmf.uni-lj.si/pub/networks/default.htm

We have used .Net graph file format to be able to use Pajek (see Appendix I).
(De Nooy et al., 2005).

3.3.3. Gephi

Gephi is a tool that can be used to examine and analyze graphs. The user can interact with the representation; during network analysis the user can make hypothesis, intuitively discover patterns, and isolate structure singularities or faults. Some of the properties of Gephi:

- Networks up to 50,000 nodes and 1,000,000 edges can be examined.
- Dynamic filtering can be used
- Provides tools for meaningful graph manipulation

3.3.3.1. Applications of Gephi

Gephi can be used for real time exploratory data analysis, link analysis to reveal the associations between the objects composing the graph, Social network analysis to discover communities, Biological network analysis to represent the patterns hidden in the biological data, and poster creation

Clustering coefficient, modularity, path length, density, diameter, centrality, degree (power-law), betweenness, and closeness metrics can be used in Gephi.

3.3.3.2. Underlying Technology

NetBeans UI that includes built-in 3D rendering engine is used to provide ergonomic interface for usage.

These formats are supported in Gephi: NET (Pajek), (GUESS), GraphML (NodeXL), GML, NET (Pajek), GEXF and more.

4. AN APPLICATION OF COMMUNITY DISCOVERY IN SOCIAL NETWORKS

As described in previous chapters there are many methods for community discovery. In this part of thesis we applied k-core community discovery method on DBLP dataset. We have used Pajek Social Network Analysis tool for community discovery.

4.1. K-core Community Discovery Process

K-core community discovery process consists of five phases. In the first phase we prepare datasets for Pajek. Pajek accepts social network in a special format called .NET .Therefore we have converted DBLP dataset into the .NET format. In the second phase, we discover k-cores and in the third phase we discover weak components. In the fourth phase we visualize the graph to prepare for the next iterations. In fifth phase we generate a report that includes several metrics. The entire process is shown in Figure 4.1

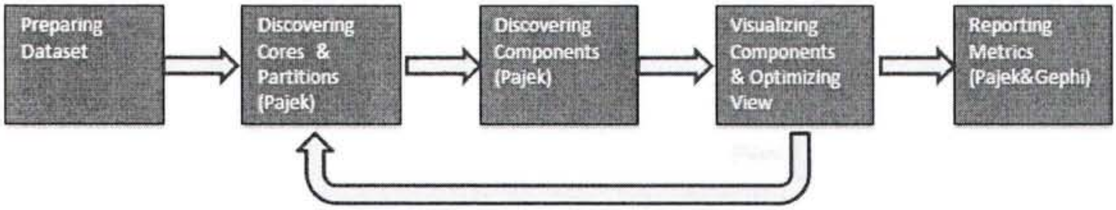


Figure 4.1 Brief representation of the framework

4.2. Data Sets

We have discovered communities using 2 different datasets. These sets are called DBLP (A Computer Science Bibliography) and Arxiv Physics network.

4.2.1. DBLP

In this thesis, DBLP data is used as the first dataset to discover the communities. DBLP is a cooperation project server that provides bibliographic information on major computer science journals and proceedings. This database is developed by Schloss Dagstuhl and Trier University (http://www.dagstuhl.de/en/about-dagstuhl/projects/lzi-dblp/). There are more than 2 million documents listed in this database. The data can be retrieved in XML format. The following shows XML document type definitions (DTD) of categories and entities in DBLP database.

```
<!ELEMENT dblp (article|inproceedings|proceedings|book|incollection|
                phdthesis|mastersthesis|www)*>
<!ENTITY % field
"author|editor|title|booktitle|pages|year|address|journal|volume|number|month|url|ee
|cdrom|cite|publisher|note|crossref|isbn|series|school|chapter">
```

We are directly interested in 'article' category and 'author' entity. This is because the relations can be extracted by analyzing these entities. The size of the entire database is about 1GB (185 MB as zipped). It is very difficult to work on entire DBLP database. Therefore we have worked on a smaller sample that is extracted from DBLP database. The obtained sample dataset includes 30 MB's of data that contains 87,555 nodes and 217,455 edges.

The entire dataset is downloaded from http://dblp.uni-trier.de/xml/ . The downloaded file is called dblp.xml.gz .

4.2.2. Arxiv high energy physics theory citation network.

In this thesis, Arxiv HEP-TH (high energy physics theory) citation dataset is used as the second dataset to discover the communities. This dataset covers all the citations within physics community that has 27,770 nodes and 379,563 edges. If a paper i cites paper j, the graph contains a directed edge from i to j. The dataset includes papers in the period from January 1993 to April 2003 (124 months).

The dataset size is nearly 4,3 MBs and it is in .NET file format that can be downloaded from:

http://snap.stanford.edu/data/cit-HepTh.html

4.3. Data Preprocessing and Conversion

DBLP dataset is stored in XML format. In this thesis, we have preprocessed this dataset and then converted into .NET network file format for our analysis. We have developed a program for data preprocessing and conversion.

4.3.1. Requirements for Data Preprocessing and Conversion

In order to complete preprocessing and data conversion process, the following installations are required:

- Cygwin must be installed
- XML to CSV parser must be installed
- Microsoft Visual C++ 2010 Express must be installed
- TXT2PAJEK must be installed
- Microsoft Excel 2007 must be installed
- Notepad ++ must be installed

Cygwin is a freeware Linux emulator that can be downloaded from http://www.cygwin.com/. Here, it is used for unzipping and splitting the DBLP .gz file because of performance reasons.

XML to CSV Convertor is freeware software developed in C# 4.0 and it can be downloaded from http://xmltocsv.codeplex.com/. Here it is used for converting DBLP XML file to category CSV file.

Microsoft Visual C++ 2010 Express is a freeware development suit, provided by Microsoft, and can be downloaded from http://www.microsoft.com/visualstudio/en-us/products/2010-editions/express.

Here, it is used for building and debugging the C++ code (See Appendix II) used for building the network relations from the CSV output of XML to CSV Convertor.

TXT2PAJEK is a freeware software used for creating .NET network file from the output of C++ conversion code described above and it can be downloaded from http://vlado.fmf.uni-lj.si/pub/networks/pajek/howto/text2pajek.htm . The . NET Pajek format is explained in Appendix I .

Notepad ++ is a freeware text editor tool used for visualizing and processing large files manually and it can be downloaded from http://notepad-plus-plus.org/.

4.3.2. Data Preprocessing Phases

The phases of the data preprocessing process for converting DBLP dataset into .NET file format is shown in Figure 4.2 :

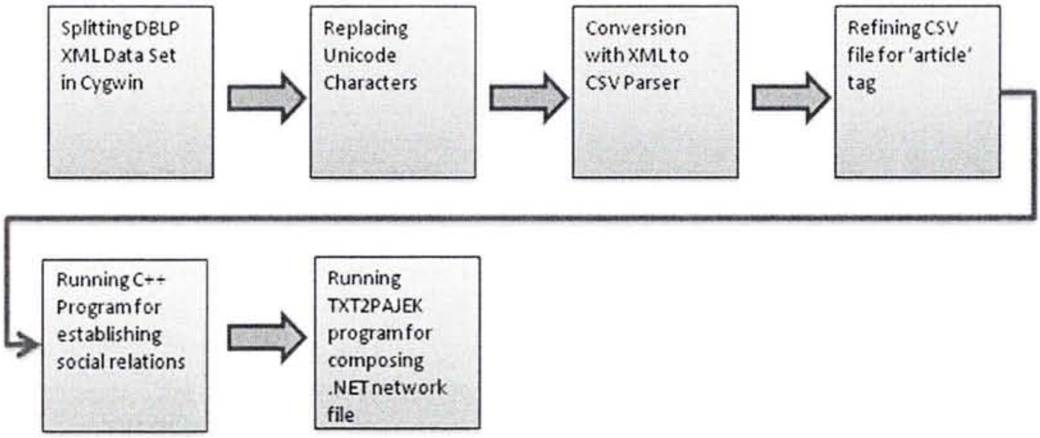


Figure 4.2 Dataset Conversion Framework

As the first step of the process, 1 GB sized DBLP data is split to 30 MB size partitions by using Cygwin Linux emulator. This is required because data conversion may cause some memory and performance problems if the file is not in a suitable size. In this thesis, we have used a subsample of DBLP dataset for producing the necessary social relations of the authors in article category. This subsample is about 30 Mbs.

Some of the special characters in some languages resulted in conversion problems. In order to solve this problem, non-ASCII characters in XML files is converted into Unicode format by using a table.

In the third step, we have used XML to CSV convertor program to extract the article – author category-entity information from the XML file. A sample run of the convertor program is shown in Figure 4.3 :

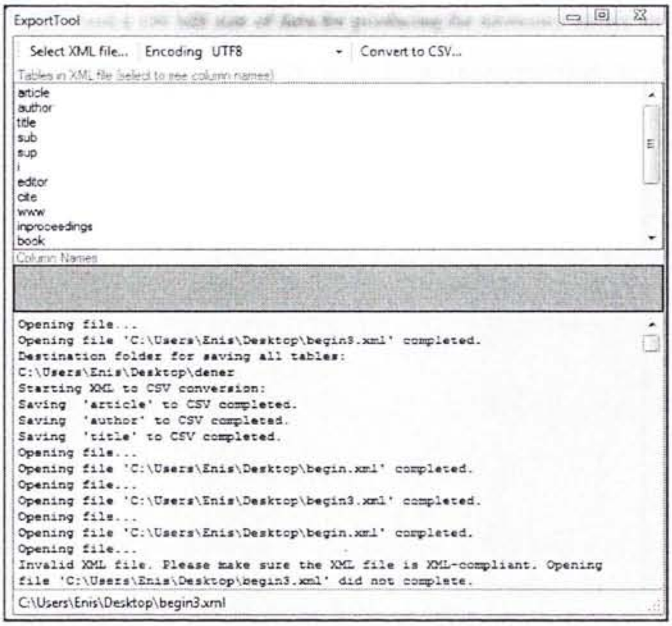


Figure 4.3 XML to .Net Convertor

This program enables the extraction of XML elements defined in tags. At the end the program processes XML file and produces a CSV file for each tag. In the fourth step, the CSV file is filtered manually in Excel to only display the relationships for the article category. The sample output is like:

author_Text;article_Id

E. F. Codd;0

Patrick A. V. Hall;1

Markus Tresch;2

E. F. Codd;3

C. J. Date;3

E. F. Codd;4

E. F. Codd;5

E. F. Codd;6

E. F. Codd;7

Michael Ley;8

Rita Ley;9

Markus Casper;9

This CSV file includes two columns separated by a semicolon. The first column is the author of the paper and the second column is the paper id that is published by the referred author. If the paper ids are the same that means that these articles are written by the corresponding authors together.

The papers written by 2 or more authors must be represented by edges. Two ends of the edges are authors that write the same article. This is required for Pajek conversion utility. For this purpose we produced an output file that stores pairs of authors that write the same article.

The sample of the output is like:

Foad Oloumi	Faraz Oloumi
Foad Oloumi	Rangaraj M. Rangayyan
Faraz Oloumi	Rangaraj M. Rangayyan

In the sixth step, we run TXT2PAJEK program for composing .NET Pajek network file.

In Pajek .NET file format, the first part includes node numbers with their descriptions. The second part of the file describes edges among nodes.

Arxiv Dataset is in .NET format so it can be used without extra data processing.

4.4. Discovering Communities in the Dataset

In this thesis, we start our analysis using Pajek version 2.05. We have used two different datasets which are stored in files that are in .NET file format obtained after the preprocessing phase. Our aim is to discover communities in these network files. The first dataset is the DBLP Dataset. DBLP dataset is in XML format and is converted to .NET file format during preprocessing phase. The second dataset is Arxiv High-energy physics theory citation network, Arxiv dataset is in .NET file format and can be directly processed by Pajek tool.

4.4.1. Characteristics of Datasets

The first dataset, DBLP Dataset, has the following characteristics:

Nodes : 87,555

Edges : 217,455

Type : Directed

Average clustering coefficient: 0.672

Diameter : 30

The second dataset is Arxiv High-energy physics theory citation network

Nodes : 27,770

Edges : 379, 563

Type : Directed

Average clustering coefficient: 0.3295

Diameter : 14

4.4.2. Analysis of DBLP Dataset

As described in 3.2.2.6. we have used K-core Community Discovery Method to discover communities. The iterations applied in Pajek for the DBLP dataset is like:

Iteration	# Nodes	#Edges	Density	Avg. Degree	Operation	# of Components	# of nodes in Largest Component
1	87.555	217.415	0,00002836	4,9664	Initial	-	-
2	87.555	171.747	0,00004971	4,3521	Symmetrize	-	-
3	46.384	132.849	0,00012349	5,7282	Weak Component	2	46384

					1000-*		
4	40.988	127.453	0,00015173	6,2190	Core 2-*	-	-
5	40.988	127.453	0,00029197	4,5031	Weak Component 1000-*	1	40.988
6	30.561	109.956	0,00023546	7,1958	Core 3-*	-	-
7	29.690	107.997	0,00024503	7,2750	Weak Component 1000-*	1	29.690
8	19.688	85.014	0,00043864	8,6361	Core 4-*	-	-
9	18.433	81.405	0,00047916	8,8325	Weak Component 1000-*	2	18.433
10	11.956	62.559	0,00087527	10,4649	Core 5-*	-	-
11	10.831	58.684	0,00100047	10,8363	Weak Component 1000-*	2	10.831
12	7.200	46.139	0,00205428	12,8164	Core 6-*	-	-
13	6.498	43.371	0,02169200	13,3490	Weak Component 1000-*	2	6.498
14	4.594	35.985	0,00341002	15,6661	Core 7-*	-	-
15	3.640	43.371	0,00473343	17,2302	Weak Component 1000-*	2	3.640
16	2.771	27.498	0,00716226	19,8470	Core 8-*	-	-
17	2.529	26.144	0,00817516	20,6754	Weak Component 1000-*	2	2.529
18	2.157	24.190	0,01039816	22,4293	Core 9-*	-	-
19	2062	23.641	0,01112012	22,9302	Weak Component 1000-*	2	2062
20	1.704	21.582	0,01486526	25,3310	Core 10-*	-	-
21	1.641	21.203	0,01574708	25,8416	Weak Component 1000-*	2	1641
22	1.408	19.678	0,01985156	27,9517	Core 11-*	-	-
23	1.339	19.128	0,02133669	28,5706	Weak Component 1000-*	2	1339
24	1.164	17.857	0,02635848	30,6821	Core 12-*	-	-
25	1.151	17.779	0,02683950	30,8931	Weak Component 1000-*	2	1151
26	1.002	16.699	0,03326381	33,2638	Core 13-*	3	548
27	934	16.200	0,03713965	34,6895	Core 14-*	3	507
28	838	15.458	0,04402316	36,8926	Core 15-*	4	465

Figure 4.4 DBLP Iterations

We have analyzed the characteristics of DBLP dataset by using Gephi tool. Analysis results for DBLP dataset is listed below:

Metric	Before the analysis:	After the analysis:
Network Type	Undirected	Undirected
Symettrized	No	Yes
K-core level	1	15
Number of vertices	87,555	838
Number of edges	217,455	15,458
Density	0.00002836	0.044023
Average Degree	4.352	36,8926
Number of weak components	10,954	4
Size of the largest component	46,384 vertices(52,977%)	465 vertices
Diameter	30	14
Average Path length	9.919	5.169
Average Clustering Coefficient	0.672	0.964
Average Embeddedness	7.074	45.854

Where

- Symmettrized means to change a network from directed to undirected.

- k-core is a maximal subset of vertices such that each is connected to at least k others in the subset

-Density is the number of ties in ratio to the total number of possible ties.

-Average degree is the average of all nodes' degree values

-Weak component can be described as if two vertices are connected by one or more paths through the network.

- Diameter of a network is defined as the longest of calculated shortest paths in a network.

-Average path length is the average of all possible paths in the network

- The clustering coefficient is defined as probability that two randomly selected neighbors are connected to each other. Average clustering coefficient is the average for all nodes.
- Embeddedness is the likelihood of a triplet being closed by a tie so that it forms a triangle.

.

After the analysis, Pajek output in Figure 4.5 shows the high-level k-cores and weak components:

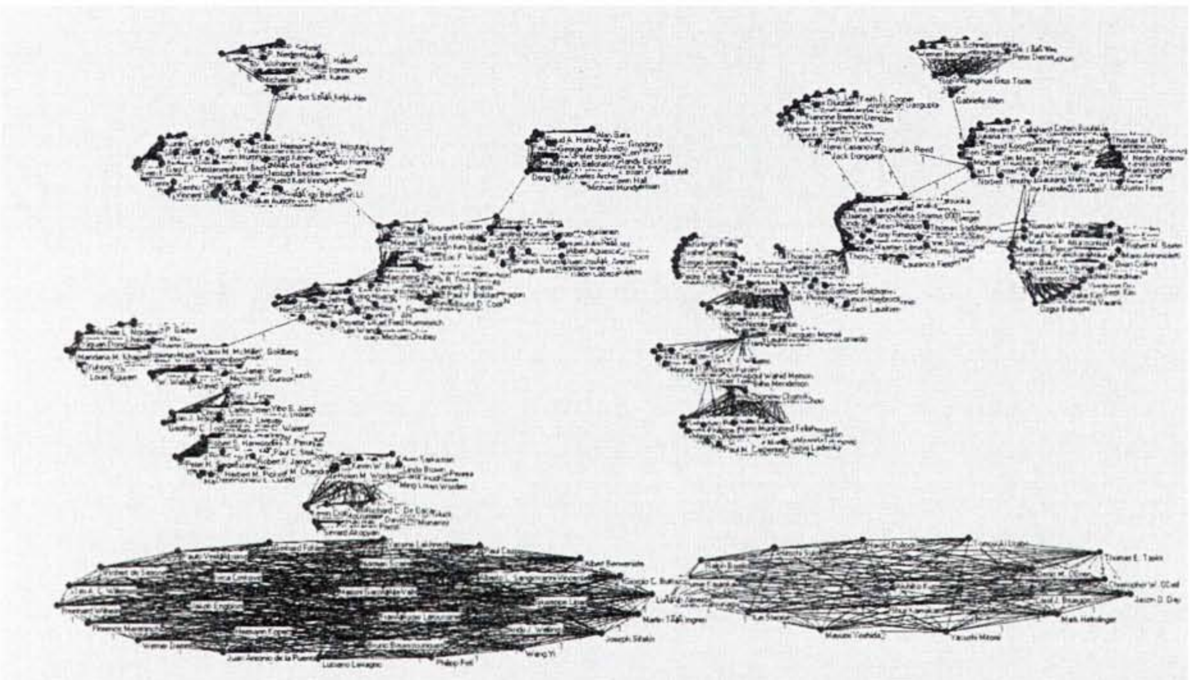


Figure 4.5 K cores and weak components of DBLP

Betwenness Centrality measures how often a node appears on shortest paths between nodes in the network. Below in Figure 4.6 and Figure 4.7 the betweenness centrality distribution of DBLP dataset is shown:

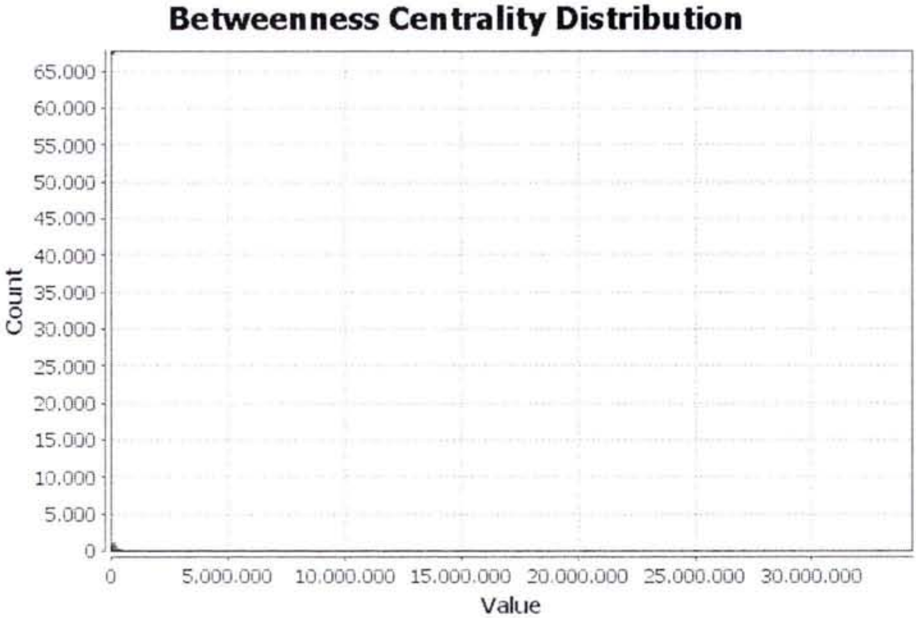


Figure 4.6 Betweenness Centrality Distribution of DBLP (Before)

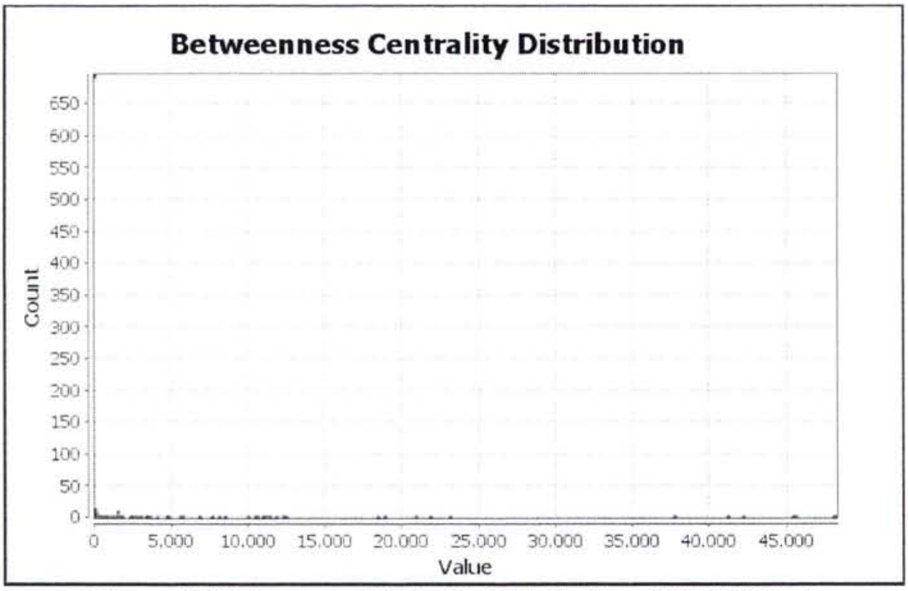


Figure 4.7 Betweenness Centrality Distribution of DBLP (after)

Closeness Centrality is the average distance from a given starting node to all other nodes in the network. Below in Figure 4.8 and Figure 4.9 the closeness centrality distribution of DBLP dataset is shown:

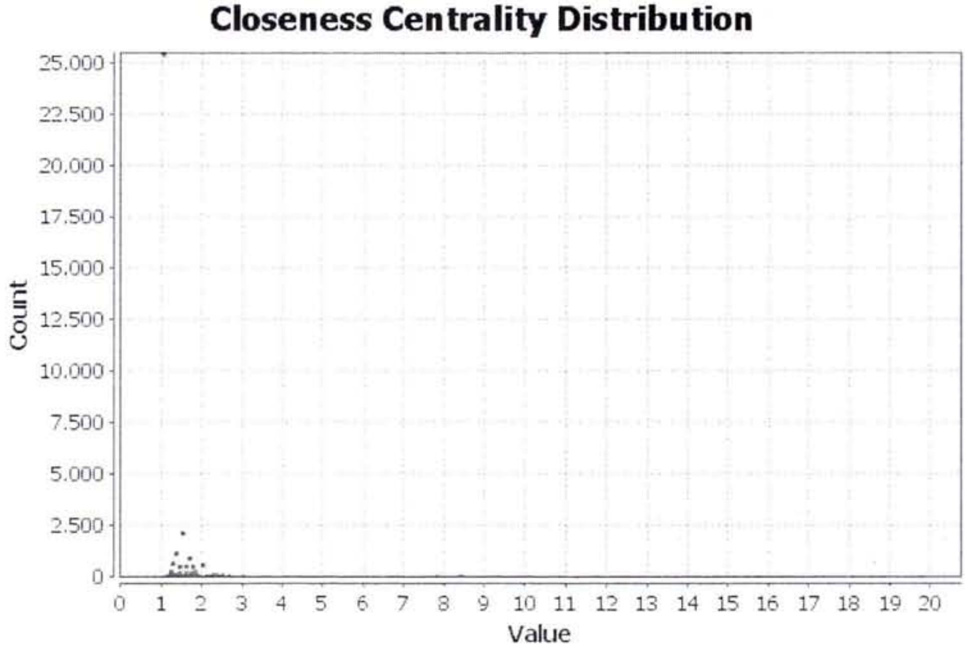


Figure 4.8 Closeness Centrality Distribution of DBLP (Before)

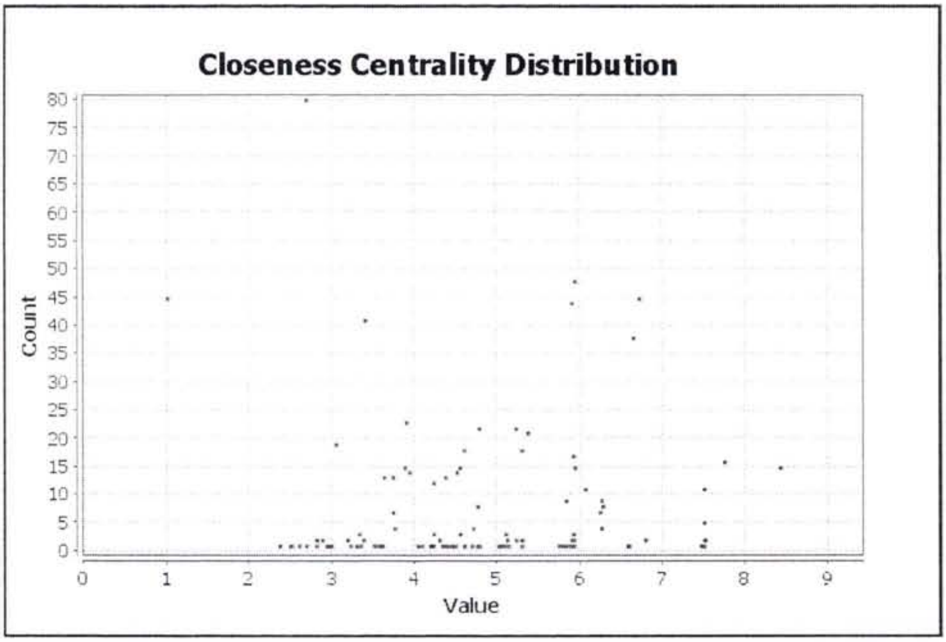


Figure 4.9 Closeness Centrality Distribution of DBLP (after)

The clustering coefficient is defined as probability that two randomly selected neighbors are connected to each other. Below in Figure 4.10 and Figure 4.11 the clustering coefficient distribution of DBLP dataset is shown:

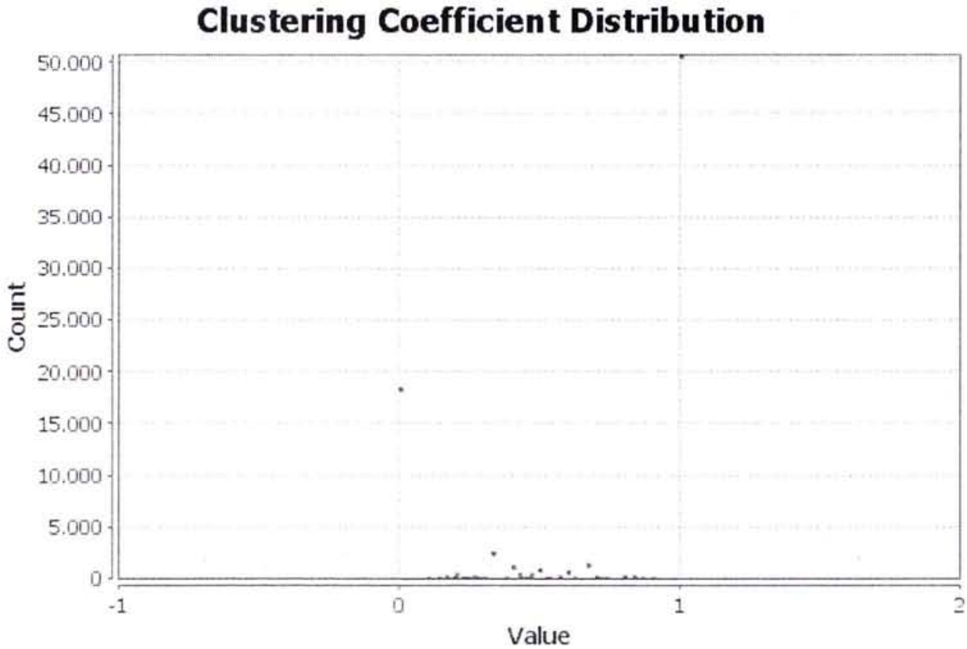


Figure 4.10 Clustering Coefficient Distribution of DBLP (Before)

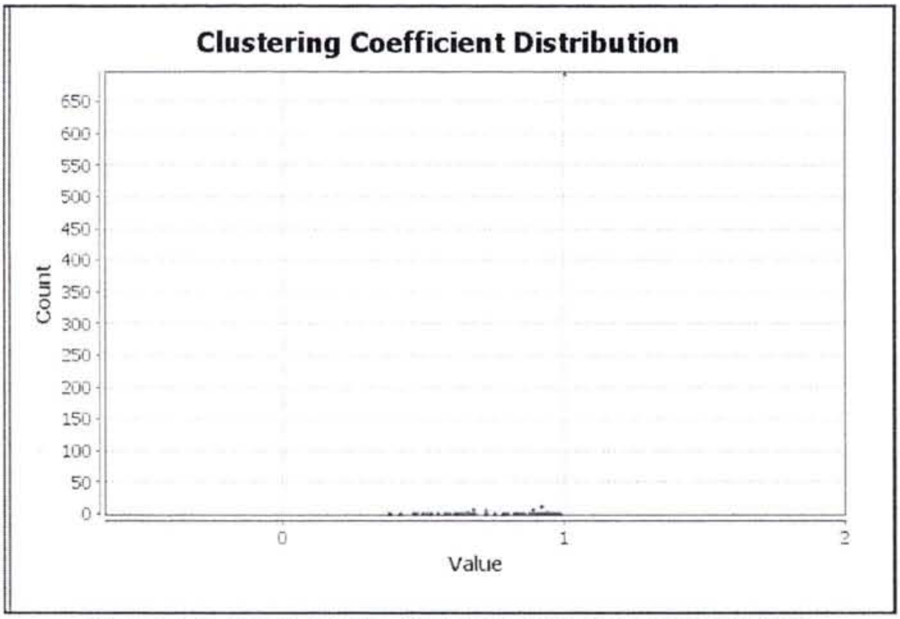


Figure 4.11 Clustering Coefficient Distribution of DBLP (after)

The resulting parameters of the clusters shown in Figure 4.5 are listed below in Figure 4.12
:

Frequency distribution of cluster values:

Cluster	Freq	Freq%	CumFreq	CumFreq%	Representative
15	42	5.0119	42	5.0119	Yuri Knyazikhin
16	51	6.0859	93	11.0979	Kei Shiomi
17	92	10.9785	185	22.0764	Eli J. Mlawer
18	32	3.8186	217	25.8950	David F. Young
19	4	0.4773	221	26.3723	Sergio Jimenez
20	19	2.2673	240	28.6396	Thomas J. Jackson
21	21	2.5060	261	31.1456	Qin Li
22	21	2.5060	282	33.6516	Gianpaolo Zanier
23	46	5.4893	328	39.1408	Hartmut Aumann
24	50	5.9666	378	45.1074	Manuel Martin-Neira
25	26	3.1026	404	48.2100	Walt C. Oechel
26	27	3.2220	431	51.4320	Ayal Zaks
27	46	5.4893	477	56.9212	Eric J. Fetzer
29	30	3.5800	507	60.5012	G. Arturo Sanchez-Azofeifa
45	46	5.4893	553	65.9905	Dong Chen
47	48	5.7279	601	71.7184	Jonathan H. Jiang
49	98	11.6945	699	83.4129	Paul T. Groth
53	54	6.4439	753	89.8568	Gary E. Christensen
84	85	10.1432	838	100.0000	Arie Shoshani
Sum	838	100.0000			

Figure 4.12 Frequency distributions of DBLP communities

As expected, after the analysis of DBLP data, the Density, Average Degree, Average Clustering Coefficient, Average Embeddedness values are increased while Diameter, Number of weak components, Number of shortest paths, Average Path length values are decreasing.

After the iterations, as in Figure 4.11, Clustering Coefficient values seem to take values near to 1 compared to the before iteration status in Figure 4.10. This leads the idea that nodes in the cluster tend to form closer groups. Closeness centrality graph in the Figure 4.8 indicates that very high number of nodes can be accepted as central, denser between the values of 1 and 2.Figure 4.9. is very similar to the the Figure 4.8 except application of weak component and k-cores transformation has removed the less central nodes from the network. Expectedly Betweenness graphs in Figure 4.6 and Figure 4.7 have similar graph structure except that values in Figure 4.7 are smaller in proportion to shrinking the graph by iterations.

4.4.3. Analysis of Arxiv Dataset

As described in 3.2.2.7 we have used Main Path Analysis Method to discover communities in Arxiv citation dataset.

We have analyzed the characteristics of Arxiv dataset by using Gephi tool. Analysis results for Arxiv dataset is listed below:

Metric	Before the analysis:	After the analysis:
Network Type	Directed	Directed
Symettrized	No	No
Number of vertices	27,770	74
Number of edges	379,563	73
Density	0.00045618	0.01333090
Average Degree	12.668	1.97297297
Number of weak components	143	1
Number of strong components	20,094	-
Size of the largest component	7,463 vertices	74 vertices
Diameter	37	68
Average Path length	8.473	24.843
Average Clustering Coefficient	0.156	0
Average Embeddedness	12.398	0
Number of shortest paths	224,125,973	2,692

Where

- Symmettrized means to change a network from directed to undirected.

- k-core is a maximal subset of vertices such that each is connected to at least k others in the subset

-Density is the number of ties in ratio to the total number of possible ties.

-Average degree is the average of all nodes' degree values

-Weak component can be described as if two vertices are connected by one or more paths through the network.

- Diameter of a network is defined as the longest of calculated shortest paths in a network.

-Average path length is the average of all possible paths in the network

- The clustering coefficient is defined as probability that two randomly selected neighbors are connected to each other. Average clustering coefficient is the average for all nodes.

- Embeddedness is the likelihood of a triplet being closed by a tie so that it forms a triangle.

- A shortest path is a path between two vertices (or nodes) in a graph such that the sum of the weights of its constituent edges is minimized.

The iterations applied in Pajek for the Arxiv dataset are listed in Figure 4.13 below:

Iteration	# Nodes	#Edges	Density	Avg. Degree	Operation	Componen t Size	Largest Componen t
1	27,770	379,563	0.00045618	25.3360	Initial	-	-
2	27,770	350.825	0.00091105	25.3013	Strong Component (level 2)	37	7,464
3	20,086	130608	0.00032373	13.0048790 2	Shrink Network	-	-
4	20,086	130469	0. 00032339	26.8060	Remove Loops	-	-
5	74	73	0. 01333090	1.97297297	SPC(Search Path Count) Applied	-	-

Figure 4.13 Main path analysis iterations in Pajek

It is a necessity for the network to be acyclic to apply SPC technique. Because of this, in steps 2,3,4 we removed the loops from the network. In step 5. SPC is applied to discover the main path. As normalization, we have chosen "Logarithmic Weights" because of the very high values obtained at the first trial. The results of the SPC are listed in Figure 4.14.

6. Main path SPC [logs] of N4 (74)

Lowest value of line: 0.49404357
Highest value of line: 0.99561910

Line Values			Frequency	Freq%	CumFreq	CumFreq%
(	...	0.4940]	0	0.0000	0	0.0000
(	0.4940 ...	0.6612]	27	36.9863	27	36.9863
(	0.6612 ...	0.8284]	20	27.3973	47	64.3836
(	0.8284 ...	0.9956]	26	35.6164	73	100.0000
Total			73	100.0000		

Figure 4.14 SPC result values

After the main path analysis, Pajek output in Figure 4.15 shows the evolution of one of the topics in Arxiv dataset, listed in Figure 4.17 .

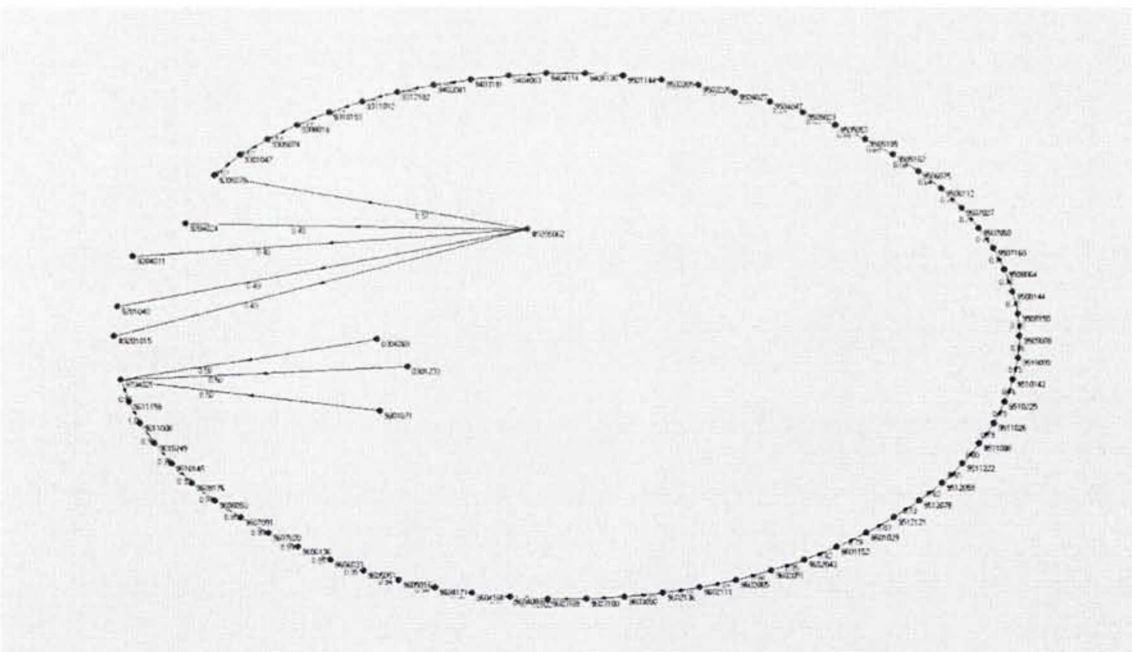


Figure 4.15 Main citation path of Arxiv Dataset

As seen in Figure 4.16, there is a 74 vertices community in the Arxiv Dataset:

```
---------------------------------------------------------------------------
3. Vertices on Main path SPC [logs] of N4 (20086)
---------------------------------------------------------------------------
Dimension: 20086
The lowest value:  0
The highest value: 1

Frequency distribution of cluster values:

   Cluster       Freq      Freq%   CumFreq  CumFreq% Representative
  ---------------------------------------------------------------
         0      20012    99.6316     20012   99.6316          1
         1         74     0.3684     20086  100.0000         65
  ---------------------------------------------------------------
       Sum      20086   100.0000
```

Figure 4.16 Community with 74 vertices of Arxiv Dataset

The result of the application of main path analysis to Arxiv Dataset: We have applied SPC (Search Path Count) method to calculate the traversal weights. We have discovered a cohesive group (community) who has published 74 articles citing each other including the similar keywords listed in Figure 4.17. These keywords are retrieved from the title and abstracts of the articles listed in the main path. See Appendix III.

String	**Brane/Symmettry**	**Other keywords**
$2+D$ - dimensional string	p-brane	Hawking Beckenstein entropy
six-dimensional string theories.	p-branes	black hole
string field theory	brane models	WZW models
string theory	D-brane	U-duality
D-string	p-branes	BPS
String	D=11 supersymmetric multiplets	Calabi--Yau threefold
string duality	duality symmetry	Entropy
string model	kappa-symmetric	F theory
string theory	Supersymmetry	five-dimensional
heterotic string duality in	Symmetry	gauge theory
self-dual strings		M-theory
superstring		supergravity.
supersymmetric string theories		WZNW models
type II strings		Yang-Mills theory
type IIA string		
three dimensional black string		
type-IIA superstring		

Figure 4.17 Common words that appears in the titles and abstracts of the papers

This 74 membered article network shows the evolution of the research traditions about the mostly used keywords in Figure 4.17. 'String', 'brane', 'symmetry' and their derivatives are the mostly seen keywords. The most influential writers are the most popular members of this research tradition, listed in Figure 4.18. The research tradition begins in January 1992 and ends in 2003. This date interval is the interval of the Arxiv dataset described in section 4.2.2

Author	Count of Author
C. Vafa	13
A.A. Tseytlin	7
Ashoke Sen	4
K. Sfetsos	4
A. Strominger	3
Edward Witten	3
Mirjam Cvetic	3
Sergio Ferrara	3
Alexander von Gussich	2
Alok Kumar	2

Figure 4.18 Most popular authors found in research tradition

5. CONCLUSION

In this thesis, we have researched the Community Detection Algorithms and methods that discover the communities in the social networks, and applied two different methods on two different datasets. We have selected two datasets: DBLP and Arxiv citation network datasets. DBLP is a cooperation project server that provides bibliographic information on major computer science journals and proceedings. Arxiv HEP-TH is a high energy physics theory citation network. For DBLP we have developed preprocessing method to convert the XML data to .NET network file format, in order to be able to process with Pajek tool.

We have applied K-core Community Discovery method to DBLP dataset to discover the change in the network characteristics and the communities. The k-core method helps discovering the communities in a network by identifying relatively the dense subnetworks inside the network. On the Arxiv dataset we have applied Main Path Analysis using Pajek to discover the main research traditions. The idea is that most important citations form one or more main paths of a research tradition. Main path analysis achieves this by calculating the traversal weight of a citation. Detected groups and communities are given and discussed in the thesis.

As future work, main path analysis can be applied to different citation datasets in the same study and by discovering the main paths; the interdisciplinary connections can be discovered.

REFERENCES

[1] Aggarwal C. C., (2011), "An Introduction to Social Network Data Analytics" , Springer Science+Business Media

[2] Newman M.E.J., (2011), "Networks An Introduction", Oxford University Press

[3] Newman M.E.J. and Girvan M., (2004), "Finding and evaluating community structure in networks" , Phys. Rev. E, 69 (2):026113

[4] Satuluri V. and Parthasarathy S., (2009), "Scalable graph clustering using stochasitc flows: applications to community discovery In KDD '09, " , ACM, 2009, 737-746

[5] Granovetter M., (1985), "Economic action and social structure: The problem of embeddedness", American Journal of Sociology, 91(3):481–510

[6] Watts D.J. and Strogatz S.H., (1998), "Collective dynamics of 'small-world'networks. Nature", pages 440–442, Jun 1998

[7] Katz L., (1953), "A new index derived from sociometric data analysis", Psychometrika,18:39–43

[8] Freeman L.C., (1979), "Centrality in social networks: Conceptual clarification. Social Networks", 1:215 239

[9] Ruef M. 1, Aldrich H.E. 2, Carter N.M. 3, (2003), "The Structure of Founding Teams: Homophily, Strong Ties, and Isolation among U.S. Entrepreneurs", American Sociological Review, Vol. 68, No. 2 (Apr., 2003), pp. 195-222.

[10] Von Luxburg U., (2007), "A tutorial on spectral clustering. Statistics and Computing", 17(4):395–416

[11] De Nooy W. 1, Mrvar A. 2, Batagelj V. 3, (2005), "Exploratory Network Analysis with Pajek", Cambridge University Press

[12] Milgram S, (1967), "The Small World Problem", Psychology Today, vol. 1, no. 1, May 1967, pp61-67

[13] Liu B, (2007), "Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data", Springer

[14] Kernighan B. W. and Lin S., (1970), "An efficient heuristic procedure for partitioning graphs", The Bell System Technical Journal, 49(2):291-307

[15] Girvan M. and Newman M. E. J, (2002), "Community structure in social and biological networks", Proc. Natl. Acad. Sci. USA 99, 7821–7826

[16] Kumar R. 1, Novak J. 2, Raghavan P. 3, Tomkins A. 4, (2003), "On the bursty evolution of blogspace",In Proceedings of the Twelfth International WWW Conference, pages 568–576

[17] Adamic L. A., (1999), "The small world web", In Proceedings of the third European Conferenceon Research and Advanced Technology for Digital Libraries, ECDL, number 1696, pages 443–452.

[18] Gibson D 1. Kleinberg J. M. 2, Raghavan P 3, (1998)," Inferring web communities from link topology", In UK Conference on Hypertext, pages 225–234

[19] Mutton P., (2004), "Inferring and visualizing social networks on internet relay chat", In Proceedings of the Eighth International Conference on Information Visualisation, pages 35–43, Washington,DC, USA, IEEE Computer Society.

[20] Marlow C., (2004), "Audience, structure and authority in the weblog community", In 54th Annual Conference of the International Communications Association, New Orleans, LA

[21] Richter Y.1, Yom-Tov E. 2, Slonim N. 3, (2010), "Predicting Customer Churn in Mobile Networks Through Analysis of Social Groups", In Proceedings of SDM,732-741.

[22] Dasgupta K.1, Singh R. 2., Viswanathan B. 3., Chakraborty D. 4., Mukherjea 5., Nanavati A. A. 6, (2008), "Social Ties and their Relevance to Churn in Mobile Telecom Networks", IBM India Research Lab

[23] http://parlab.eecs.berkeley.edu/wiki/_media/patterns/graph_partitioning.pdf

[24] http://en.wikipedia.org/wiki/Kernighan-Lin_algorithm

[25] http://en.wikipedia.org/wiki/Graph_partition)

[26] http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/hierarchical.html

APPENDIX I. .NET PAJEK NETWORK FILE SAMPLE

(http://www.ccsr.ac.uk/methods/publications/snacourse/netdata.html)

A Pajek network file is a simple text file which you can write in Notepad, TextPad or WinEdt for instance. It must be a simple text file; Microsoft™ Word, for example, often gives unseen character which makes the text file not simple at all, so it is to be avoided. Give it extension .net and save it in a directory say c:\temp and take note of this directory. At the end of this you should have a file in your directory c:\temp\hawthorne-friend.net which you can work on later.

Clearly Pajek network data closely followed Graph theory concepts described above because it has two parts (each marked by asterisk). The first part after the comments must start with *Vertices, that is asterisk and Vertices without a space between them, and the number of vertices. It is followed by a sequence of integers and labels, where the integer sequence starts from 1. The labels are one word or more, in which case it has to be double quoted like: 1 "Inspector 1". The second part is marked by *Edges, that is an asterisk and Edges without a space between them, which is followed by edges list, and ends with a blank line. Please include the last and only one blank line, i.e. your cursor must be on that blank line when you save the file, otherwise Pajek will be confused.

```
/* Cut from this line to the last line                 */
/* Save this in your local directory, say C:\temp      */
/* give it a name:                                     */
/* hawthorne-friend.net                                */
/* Of course you can just download this data from the  */
/*      link above by right clicking and saving it     */
/*                                                     */
/* Source: Roethlisberger and Dickson 1939 :501ff      */
/*                                                     */
/* An example of non directed or simple social network */
/* It has two parts (each marked by asterisk) which    */
/*   closely followed Graph theory definition of graph */

*Vertices 14
1 I1
2 I3
3 W1
4 W2
5 W3
6 W4
7 W5
8 W6
9 W7
```

```
10 W8
11 W9
12 S1
13 S2
14 S4
*Edges
1 5
3 5
3 6
5 6
9 10
9 11
10 11
3 12
5 12
6 12
10 14
11 14
9 12
```

APPENDIX II. C++ CODE OF DATASET REFINEMENT

*** social.cpp ***
*** author : Enis Arslan ***

```
# include <cstdlib>
# include <iostream>
# include <fstream>
# include <string>
#include <map>
#include <sstream>

using namespace std;

	std::string matris [100000] [2];
	ifstream aa ("C:\\users\\enis\\desktop\\19may.txt");
	int i=0 ;
	int j=0;
	int sayac;
	string x  ;
	std::string yourarray[100000][2];
	int matrislen;

	ofstream arrayData("C:\\users\\enis\\desktop\\array.txt"); // File
Creation(on C drive)

	void snaoutput (int i, int say)
	{
		int t=0;
		int son;
		int count ;

		son = say ;
		while (son>0)
		{
			count=son;
			while (count>0)
			{

				yourarray [t][0] = matris [i+son][0];
				yourarray [t][1] = matris [i+count-1][0];
				arrayData << yourarray [t][0] << '       ' <<yourarray
[t][1] << endl;
				count--;
				t++;
		}
		son--;
		}


	}

int main (int argc, char argv[])
{

  std::string line;
  std::map<int, std::string> map;
```

```
if(!aa) //Always test the file open.

  {
            cout<<"Error opening output file"<<endl;
            system("pause");
            return -1;
  }

while (std::getline(aa, line)) {
  std::string::size_type pos = line.find(';');
  std::stringstream sstream(line.substr(0, pos));
  int index;
     int index2;
  sstream >> index;
     sstream >> index2;
  map[index] = line.substr(pos+1);
     matris [i][1] =  map[index];
     map[index2] = line.substr(0, pos);
     matris [i][0] = map[index2];
     i++;

     }
     matrislen = i;
     i=0;
     while (i<matrislen)
     {
     j= i ;
     sayac = 0;

            while  (matris [j][1] ==  matris [j+1][1])
            {
                   sayac++;
                   j++;
            }

            snaoutput (i,sayac);

            i=j+1;
            j=0;
            sayac = 0;

     }


     system("PAUSE");
     return EXIT_SUCCESS;

}
```

APPENDIX III. KEYWORDS OF THE MAIN PATH ARTICLES

No	Arxiv ID	Title	Author	Month	Year	Keywords
1	304269	Strings, p-Branes and Dp-Branes With Dynamical Tension	Eduardo Guendelman, Alexander Kaganovich (Ben-Gurion Univ., Beer-Sheva), Emil Nissimov, Svetlana Pacheva	4	2003	brane models,p-brane
2	301233	Parent Actions, Dualities and New Weyl-invariant Actions of Bosonic p-branes	Yan-Gang Miao, Nobuyoshi Ohta	1	2003	p-branes ,duality symmetries
3	9801071	Vacuum energy for the supersymmetric twisted D-brane in constant electromagnetic field	A.A. Bytsenko, A.E. Goncalves, S. Nojiri, S.D. Odintsov	4	1998	D-brane
4	9704021	Intrinsic Geometry of D-Branes	M. Abou Zeid, C. M. Hull	5	1997	D-brane ,p-branes ,symmetry
5	9611159	The Dirichlet Super-p-Branes in Ten-Dimensional Type IIA and IIB Supergravity	Martin Cederwall, Alexander von Gussich, Bengt E.W. Nilsson, Per Sundell, Anders Westerberg	11	1996	p-branes,supersymmetric
6	9611008	D=11, p=5	P.S. Howe, E. Sezgin	11	1996	brane ,supersymmetry
7	9610249	D-Brane Actions with Local Kappa Symmetry	Mina Aganagic, Costin Popescu, John H. Schwarz	10	1996	brane,supersymmetry, supergravity
8	9610148	The Dirichlet Super-Three-Brane in Ten-Dimensional Type IIB Supergravity	Martin Cederwall, Alexander von Gussich, Bengt E.W. Nilsson, Anders Westerberg	10	1996	brane,supersymmetric ,kappa-symmetric,supergravity.
9	9609176	Unification of String Dualities	Ashoke Sen	9	1996	string,M- and F-theory
10	9608053	F Theory Orientifolds, M Theory Orientifolds, and Twisted Strings	Julie D. Blum	8	1996	string,F theory and M theory
11	9607091	A Non-Perturbative Superpotential With E_8 Symmetry	R. Donagi, A. Grassi, E. Witten	7	1996	symmetry
12	9607020	A Test Of The Chiral E8 Current Algebra On A 6D Non-Critical String	Ori J. Ganor	7	1996	2-branes,M-theory
13	9606136	On Tensionless Strings in $3+1$ Dimensions	Amihay Hanany, Igor R. Klebanov	6	1996	2-brane,string,M-theory ,5-branes
14	9606033	Non-extreme black holes from non-extreme intersecting M-branes	M. Cvetic, A.A. Tseytlin	6	1996	p-brane ,2-branes,string,5-branes,black holes
15	9605051	Near-BPS-Saturated Rotating Electrically Charged Black Holes as String States	Mirjam Cvetic, Donam Youm	5	1996	entropy,black holes,string

16	9605016	Statistical Entropy of Near Extremal Five-branes	Juan M. Maldacena	5	1996	5-branes,black holes,Hawking Beckenstein entropy
17	9604171	Self-Dual Superstring in Six Dimensions	John H. Schwarz	4	1996	superstring,M theory
18	9604166	Intersecting M-branes as Four-Dimensional Black Holes	I.R. Klebanov, A.A. Tseytlin	4	1996	p-brane ,supersymmetric,black holes,5-branes,2-branes,3-branes, Bekenstein-Hawking entropy
19	9604089	Entropy of Near-Extremal Black p-branes	I.R. Klebanov, A.A. Tseytlin	4	1996	\$p\$-branes,supersymmetric,black holes,5-branes,2-branes,3-branes, Bekenstein-Hawking entropy
20	9603109	Nonextremal Black Hole Microstates and U-duality	Gary Horowitz, Juan Maldacena, Andrew Strominger	3	1996	D-brane,black holes,string theory,Bekenstein-Hawking entropy
21	9603100	General Rotating Five Dimensional Black Holes of Toroidally Compactified Heterotic String	Mirjam Cvetic, Donam Youm	3	1996	superstring,black holes
22	9603090	Universality of Sypersymmetric Attractors	Sergio Ferrara, Renata Kallosh	3	1996	entropy,supersymmetric theory ,supersymmetry
23	9602136	Supersymmetry and Attractors	Sergio Ferrara, Renata Kallosh	2	1996	Bekenstein-Hawking entropy,supersymmetry,black hole
24	9602111	Macroscopic Entropy of \$N=2\$ Extremal Black Holes	Andrew Strominger	2	1996	five dimensions
25	9602065	D--branes and Spinning Black Holes	J.C. Breckenridge (1), R.C. Myers (1), A.W. Peet (2), C. Vafa (3) ((1) McGill, (2) Princeton, (3) Harvard)	2	1996	D-brane ,black hole,five dimensions
26	9602051	Counting States of Near-Extremal Black Holes	Gary Horowitz, Andrew Strominger	2	1996	Bekenstein-Hawking entropy
27	9602043	D-brane Approach to Black Hole Quantum Mechanics	Curtis G. Callan, Juan M. Maldacena	2	1996	D-brane ,black holes,entropy
28	9601152	Excitations of D-strings, Entropy and Duality	Sumit R. Das, Samir D. Mathur	1	1996	BPS,D-string
29	9601029	Microscopic Origin of the Bekenstein-Hawking Entropy	A. Strominger, C. Vafa	1	1996	BPS,Bekenstein-Hawking ,black holes,five-dimensional
30	9512121	BPS States, String Duality, and Nodal Curves on K3	Shing-Tung Yau, Eric Zaslow	12	1995	BPS,string
31	9512078	Instantons on D-branes	Cumrun Vafa	12	1995	branes,string
32	9512059	Open P-Branes	Andrew Strominger	12	1995	\$p\$-branes ,self-dual strings
33	9511222	D-Branes and Topological Field Theories	M. Bershadsky, V. Sadov, C. Vafa	11	1995	BPS,D-brane,2-branes,4-brane,string duality

34	9511088	Gas of D-Branes and Hagedorn Density of BPS States	Cumrun Vafa	11	1995	BPS,0-branes
35	9511026	U-duality and Intersecting D-branes	Ashoke Sen	11	1995	U-duality,string states,supersymmetry ,branes
36	9510225	D-Strings on D-Manifolds	M. Bershadsky, V. Sadov, C. Vafa	10	1995	D-strings ,symmetries
37	9510142	An N=2 Dual Pair and a Phase Transition	Paul S. Aspinwall	10	1995	string
38	9510093	Chains of N=2, D=4 heterotic/type II duals	G. Aldazabal, L.E. Ibanez, A. Font, F. Quevedo	10	1995	
39	9509009	Exact Monodromy Group of N=2 Heterotic Superstring	I. Antoniadis, H. Partouche	9	1995	duality symmetry ,superstring,Yang-Mills theory
40	9508155	Nonperturbative Results on the Point Particle Limit of N=2 Heterotic String Compactifications	S. Kachru, A. Klemm, W. Lerche, P. Mayr, C. Vafa	8	1995	string duality,supersymmetric
41	9508144	Type IIA-Heterotic Duals With Maximal Supersymmetry	S. Chaudhuri, D.A. Lowe	8	1995	supersymmetry
42	9508064	Dual Pairs of Type II String Compactification	Ashoke Sen, Cumrun Vafa	8	1995	supersymmetry,gauge symmetry,duality symmetry
43	9507168	N=1 String Duality	J.A. Harvey, D.A. Lowe, A. Strominger	7	1995	supersymmetry,string theory
44	9507050	Dual String Pairs With N=1 And N=2 Supersymmetry In Four Dimensions	Cumrun Vafa, Edward Witten	7	1995	superstrings
45	9507027	Type IIA Dual of the Six-Dimensional CHL Compactification	John H. Schwarz, Ashoke Sen	7	1995	string theory
46	9506112	K3--Fibrations and Heterotic-Type II String Duality	A.Klemm, W.Lerche, P.Mayr	6	1995	supersymmetric string theories
47	9506075	A Search for Non-Perturbative Dualities of Local $N=2$ Yang--Mills Theories from Calabi--Yau Threefolds	A. Ceresole, M. Billo', R. D'Auria, S. Ferrara, P. Fre', T. Regge, P. Soriani, A. Van Proeyen	6	1995	Duality symmetries,Calabi--Yau threefold
48	9505162	Second-Quantized Mirror Symmetry	S. Ferrara, J. A. Harvey, A. Strominger, C. Vafa	5	1995	supersymmetry,Calabi--Yau threefold,gauge theory
49	9505105	Exact Results for N=2 Compactifications of Heterotic Strings	Shamit Kachru, Cumrun Vafa	5	1995	string theory ,Yang-Mills theory
50	9505053	A One-Loop Test Of String Duality	Cumrun Vafa, Edward Witten	5	1995	heterotic string duality in
51	9505023	A Stringy Test of the Fate of the Conifold	Cumrun Vafa	5	1995	type II strings
52	9504047	The Heterotic String is a Soliton	J. A. Harvey, A. Strominger	4	1995	six-dimensional string theories.
53	9504027	STRING STRING DUALITY CONJECTURE IN SIX DIMENSIONS AND CHARGED SOLITONIC STRINGS	Ashoke Sen	4	1995	string duality,type IIA string

54	9503228	STRINGY EVIDENCE FOR D=11 STRUCTURE IN STRONGLY COUPLED TYPE II-A SUPERSTRING	Itzhak Bars	4	1995	type-IIA superstring ,D=11 supersymmetric multiplets
55	9503205	Ghost-Free Spectrum of a Quantum String in SL(2,R) Curved Spacetime	Itzhak Bars	3	1995	WZW models
56	9501144	Irrational Conformal Field Theory	M.B. Halpern, E. Kiritsis, N. Obers, K. Clubok	2	1995	field theory
57	9406136	All WZW Mooels in $D\leq5$	A.A. Kehagias	1	1994	symmetric,
58	9404114	Superstring Gravitational Wave Backgrounds with Spacetime Supersymmetry	E. Kiritsis, C. Kounnas, D. Luest	4	1994	type II
59	9404063	Four Dimensional Plane Wave String Solutions with Coset CFT Description	K. Sfetsos, A.A. Tseytlin	4	1994	WZW models
60	9403191	Plane Gravitational Waves in String Theory	I. Antoniadis, N.A. Obers	3	1994	String theory
61	9403041	Exact string background from a WZW model based on the Heisenberg group	A.A. Kehagias, P.A.A. Meessen	3	1994	D-dimensional string, WZW models
62	9312182	On Bosonic and Supersymmetric Current Algebras for Non-Semi-Simple Groups	N. Mohammedi	12	1993	Supersymmetric
63	9311012	Duality invariant class of exact string backgrounds	C. Klimcik, A.A.Tseytlin	11	1993	$2+D$ - dimensional string,
64	9310159	Antisymmetric tensor coupling and conformal invariance in sigma models corresponding to gauged WZNW theories	K. Sfetsos, A.A. Tseytlin	10	1993	String, WZW models
65	9308018	Chiral gauged WZNW models and heterotic string backgrounds	K.Sfetsos, A.A.Tseytlin	8	1993	String, WZW models
66	9305074	Effective Action and Exact Geometry in Chiral Gauged WZW Models	Konstadinos Sfetsos	5	1993	string model, WZW models
67	9301047	Exact Effective Action and Spacetime Geometry in Gauged WZW Models	I. Bars, K. Sfetsos	1	1993	String
68	9206078	A Closed, Expanding Universe in String Theory	Chiara R. Nappi, Edward Witten	6	1992	WZW models
69	9205062	Four Dimensional 2-Brane Solution in Chiral Gauged Wess-Zumino-Witten Model	Swapna Mahapatra	5	1992	2-brane
70	9204024	Target Space Structure of a Chiral Gauged Wess-Zumino-Witten Model	Supriya K. Kar, Alok Kumar	3	1992	three dimensional black string
71	9204011	Target Space Structure of a Chiral Gauged Wess-Zumino-Witten Model	Supriya K. Kar, Alok Kumar	3	1992	three dimensional black string
72	9201040	Target Space Duality as a Symmetry of String Field Theory	Taichiro Kugo, Barton Zwiebach	1	1992	string field theory
73	9201015	An Algorithm to Generate Classical Solutions for String Effective Action	S. Kar, S. Khastgir, A. Kumar	1	1992	black hole,string

CURRICULUM VITAE

ENİS ARSLAN

PERSONAL INFORMATION

Date of Birth	:	June 27,1976
City, Country of Birth	:	Ereğli/Zonguldak, Turkey
Citizen	:	Turkish
Marital Status	:	Single
Military Obligation	:	Completed

EDUCATION

2009- cont.	Doğuş University Master Program for Computer Engineering
2007-2009	Beykent University MBA Program
1994-2000	Yıldız Technical University Computer Engineering

EXPERIENCE

September 2012 – Cont'd Turkcell A.Ş. IT Department / İstanbul

- Senior Operation Engineer

October 2009 – August 2012 Nobel İlaç A.Ş. IT Department / İstanbul

- Business Services Project Management
- SAP System Application Responsibility
- Business Process Analyst

May 2007 – October 2008 AVEA İletişim Hizmetleri A.Ş. IT Department / İstanbul

- Fraud Reporting Analyst
- Oracle Application Expert

October 2006 – May 2007 J&J Turkey IT Department / İstanbul

- Business Analyst

April 2004 – October 2006 Digiturk A.Ş. IT Department / İstanbul

- Interactive Channels System Analyst

September 2000 – May 2001 Turkcell A.Ş. IT Department / İstanbul

- Webdesigner

Doğuş Üniversitesi Kütüphanesi
0007726