Pattern Searching in a Social Network

Hyosook Jung, Seongbin Park

Department of Computer Science Education, Korea University, Seoul, Korea,
{est0718, psb}@comedu.korea.ac.kr

Abstract: In this paper, we propose a system that allows a user to find a pattern that has a certain property in a social network. It exploits structural properties of a social network using geometric hashing that has been used in model-based recognition. Since it requires computationally intensive work, we construct a small-scale Grid environment based on JXTA technologies in order to preprocess and recognize patterns in different networks. We experimented the system with four types of data; blog data, web pages, OWL documents, and sociemetric data. In all cases, the result indicated that our system could find the nodes that match patterns correctly.

Keywords: social network, geometric hashing

1. Introduction

Social networks support formation of online communities by combining blogs, wikis, tagging, bookmarking services, and social networking software [1]. Several research works have analyzed the relationships in networks such as the link structure of web pages. The relationships show unique features of each node in the network. For example, a node that serves as an index node or a menu node may have many outgoing links to other nodes. It is also possible that an interesting node has many incoming links from other nodes. These features can be useful for users to find special nodes suited to their interests or needs.

In this paper, we present a system that allows a user to find a pattern which has a certain property in a social network. Using the system, a user can find nodes that satisfy structural constraints specified in a query from a social network. For example, if a user wants to find a node which has two outgoing links, one incoming link and one pair link in network n, the user specifies the constraints as query pattern p and our system finds patterns matched with query pattern p in network n. (Figure 1)

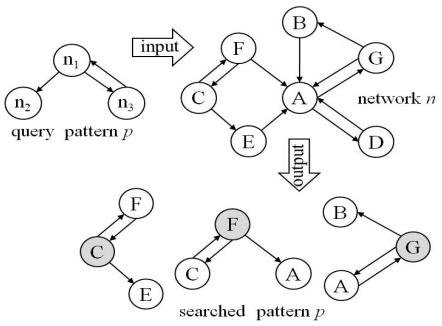

Figure 1. An example query and the results

We model a social network as a directed graph and use geometric hashing [7] that has been used in model-based recognition to search patterns that satisfy constraints. The algorithm proceeds in two steps which are preprocessing step and recognition step. In the preprocessing step, the link structure of a node is represented as a 3D geometric pattern and its invariant properties are stored into hash tables, where the invariant properties of a node are found by checking outgoing links, incoming links, and pair links of the node. In the recognition step, patterns that match against user's query are found.

The proposed system was tested experimentally under four types of social networks. First, it was applied to search communities that have unique relationship patterns on the Web. For example, some communities are linked by other communities but have a few links to other communities while others are linked by only a few other communities but have many links to other communities. Secondly, we applied it to the analysis of the link structure among web pages so that we can understand the role of each web page in the set of web pages. For example, the index page has many outgoing links to other pages. The menu or table of contents has a set of links that lead to various sections of the web site. It can help users find web pages suited to their needs. Thirdly, we applied it to the set of OWL documents which describe the classes and relations between them [10]. Fourthly, we experimented the proposed system against socio-metric data. In all cases, experimental results indicate that the proposed system finds nodes that satisfy different types of constraints specified in queries correctly.

This paper is structured as follows. Section 2 describes related works and section 3 explains the proposed approach in detail. The structure of the system is described in section 4 and illustrative examples are given in section 5. Section 6 describes experimental results and section 7 concludes the paper.

2. Related Works

Geometric Hashing is a technique for matching a target object against a set of one or more models. In the preprocessing step for each model object, it picks a reference frame and computes the basis associated with this reference frame and its shape signature. It computes the coordinates of all the other points in this reference frame and uses each coordinate as an address to the hash table. It stores the entry with model id, reference frame, shape signature, etc. at the hash table address. It repeats these tasks for each model reference frame. When performing the matching of a target object, for each reference frame of the target, it computes the basis and the shape signature associated with it and the coordinates of all other points in the current reference frame. It uses each coordinate to access the hash table and retrieves all the records such as model id, reference frame, shape signature, etc. For records with matching shape signature, it votes for the model id and the reference frame that is a potential match. For each potential match, it recovers their transformations of the potential matches and records the pairs of matching points. It verifies the match list against the target object [7].

Geometric hashing has been used in bioinformatics such as protein function determination, protein searching, structure classification and structure alignment. In [12], a JXTA-based system uses geometric hashing for protein searching, comparing protein structures. Geometric hashing can be also used for music retrieval. In [13], the system uses the geometric hashing for music information retrieval, representing a melody as a geometric structure.

The link analysis among sets of web pages is an effective method to improve the retrieval accuracy of Web search engines [4].

PageRank is a link analysis algorithm that assigns a numerical weighting to each element of a hyperlinked set of documents. It aims at measuring its relative importance within the set of documents. PageRank represents a link from Page A to Page B as a vote for Page B by Page A. It evaluates the importance of a page by the number of votes it receives. It also considers the importance of pages which casts a vote. If a page receives votes from some pages that have greater value, PageRank gives the page greater value. Important pages obtain a higher PageRank and emerge at the top of the search results [14].

HITS (Hyperlink-Induced Topic Search) is a link analysis algorithm that decides two values for a page: its authority, which estimates the value of the content of the page, and its hub value, which estimates the value of its links to other pages [2]. It first collects a set of web pages related to user's query, computes authority and hub score of each page by using power iteration method. Then, it shows the user top ranked pages as authorities and hubs based on the scores [3]. A hub is a node that links nodes with high authority values, where authority value is computed as the sum of the scaled hub values that link to a page. A hub value is the sum of the scaled authority values of the pages that the hub links to. If a node is linked by many hubs, it has high authority value. If a node links node that sis linked by many hubs, it has high authority value.

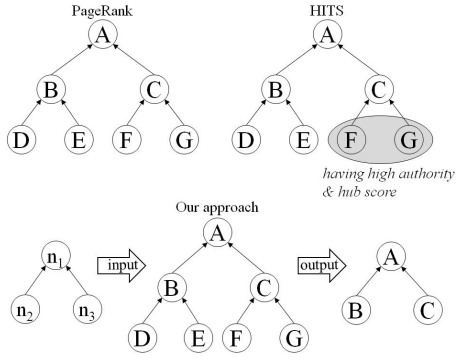

Figure 2. PageRank, HITS, and our approach

Our approach is similar to PageRank and HITS, but it considers incoming, outgoing and pair links. It searches nodes matching against a query node. Both PageRank and HITS consider incoming links and analyzing relative importance of each node. They aim at finding nodes with important information. In Figure 2, according to PageRank, node B and

C have the same rank because the numbers of incoming links are the same. According to HITS, node C will be ranked higher than node B because node C is linked by node F and G which have high authority and hub score. In our approach, node A is the only matching node because node B and C have one outgoing link. It aims at finding node with unique features rather than nodes with important information.

3. Approach

Given a social network and a query that specifies properties of a pattern, our system finds nodes that match the pattern in the network in two steps; preprocessing step and recognition step. In this section, we explain both steps in detail.

3. 1 Preprocessing step

In the preprocessing step, the information about the network under consideration is stored in a hash table. Assume that the number of nodes in the network is N. For each of the nodes in the network, the following steps (from 1 to 6) are done.

1. Count the number of in-links, out-links, and pair-links of a node and compute in-link state (I.S), out-link state (O.S), and pair-link state (P.S) for the node as follows.

I.S = I/(N−1), O.S = O/(N−1), P.S = P/(N−1)

I.S, O.S, and P.S show their positions in a network such as which node is preferred relatively to others in the network. I.S, O.S, and P.S are represented by 3D coordinates. Then, we construct a pattern for the node based on I.S(x), O.S(y) and P.S (z) as representing the three values by 3D coordinates, connecting each pair of two values, and making a segment. Figure 3 shows a geometric pattern of a node in a network, where in-link = 19, out-link = 5, pair-link = 4, N = 30, I.S = 19/29 = 0.655, O.S = 5/29=0.172, P.S = 4/29=0.138.

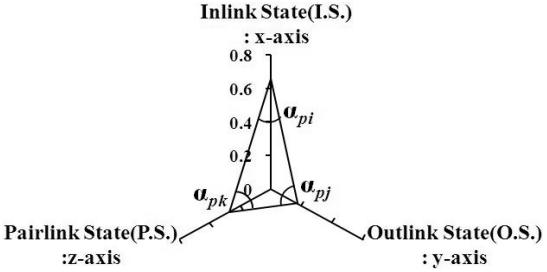

Figure 3. An example pattern

2. Calculate relative position of the node p in order to notice whether the rate of in-links is higher than that of out-links or not.

I.O.S = I.S - O.S (I.S = inlink state, O.S = outlink state, I.O.S = inlink outlink state).

3. Calculate cosines of each angle of the pattern p (cos α_{pi}, cosα_{pj}, cos α_{pk}) so that key k_p for the pattern p with the invariant properties (cos α_{pi}, cosα_{pj}, cos α_{pk}, I.O.S_p) can be computed. For the pattern given in figure 4, cos α_{pi} = 0.441, cosα_{pj} = 0.441, cos α_{pk} = 0.609, and k_p is computed as follows:

$k_p = (0.441 + 0.441 + 0.609 + 0.0333) * 10^9$

$=1.524 * 10^9$

4. Choose a hash table size m (5039) which is a prime that is not near a power of 2 and calculating a hash value h(k_p) of the pattern p with a hash function h(k_p) = k_p mod m.

h(k_p) = k_p mod m = $1.524 * 10^9$ mod 5039

= 1238

5. Insert the information about key k_p in a hash table T. An element in the hash table is connected to the invariant properties of the pattern which are stored in a database.

6. Create record r_p for pattern p whose index is h(k_p) in a database and store the invariant properties of the pattern p and other information such as the length of each segment and the values of each axis in the record r_p.

Figure 4 depicts the preprocessing step.

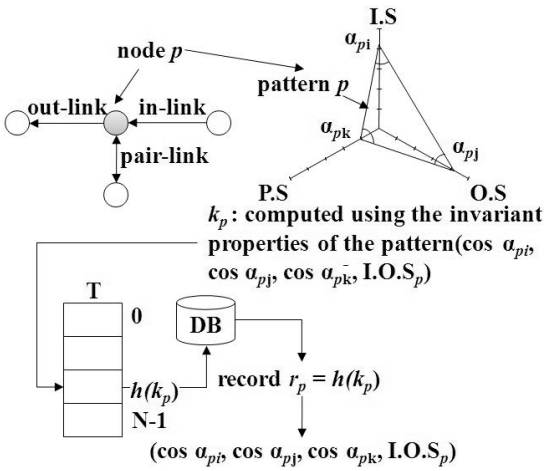

Figure 4. Preprocessing step

3. 2 Recognition step

In the recognition step, a query pattern is processed so that nodes that satisfy the constraints for the pattern can be found. Given a query pattern q, the system computes the hash value for the pattern so that nodes that match the pattern can be found.

1. Construct the query pattern q of a node by following step 1 to 2 at the preprocessing stage and calculate the hash value h(k_q) of the query pattern q by following step 3 to 4 at the preprocessing stage

2. Search candidate patterns of which the hash values are equal to that of the query pattern q. In other words, the system checks whether the hash value of query pattern q (i.e., h(k_q)) is equal to that of the pattern p (i.e., h(k_p)).

3. Compare cosines of the query node q with those of the candidate patterns. Vote matching patterns if the gap between their cosines and the cosines of the query pattern q is below the threshold t ($5.0*10^{16}$). For example, if a pattern p is a matching pattern, the values subtracting each cosine of the pattern q from each cosine of pattern p should be below the threshold (($\cos\alpha_{pi}$ - $\cos\alpha_{qi}$ <= t) and ($\cos\alpha_{pj}$ - $\cos\alpha_{qj}$ <= t) and ($\cos\alpha_{pk}$ - $\cos\alpha_{qk}$ <= t)).

Figure 5 depicts the recognition step.

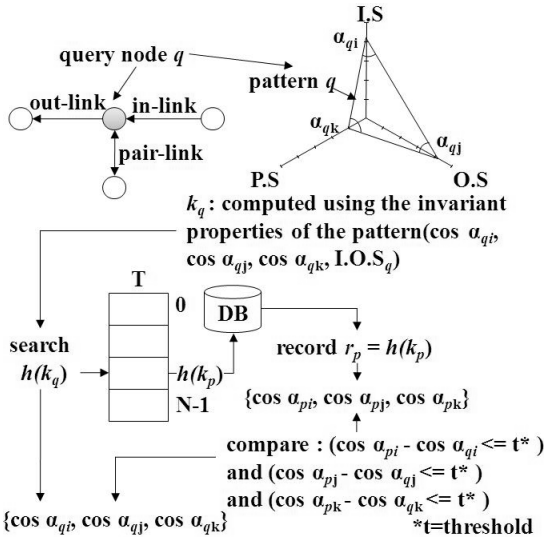

Figure 5. Recognition step

Our system can be used to find various types of patterns that exist in a network under consideration. There are two types of recognitions: general recognition and special recognition. In general recognition, a user specifies the number of inlinks, outlinks, and pairlinks of a pattern. Then the system finds nodes that satisfy the constraints. Special recognition considers nodes of specific property such as index, start, isolated, equivalent, and community node.

Figure 6 describes types of patterns that our system can find.

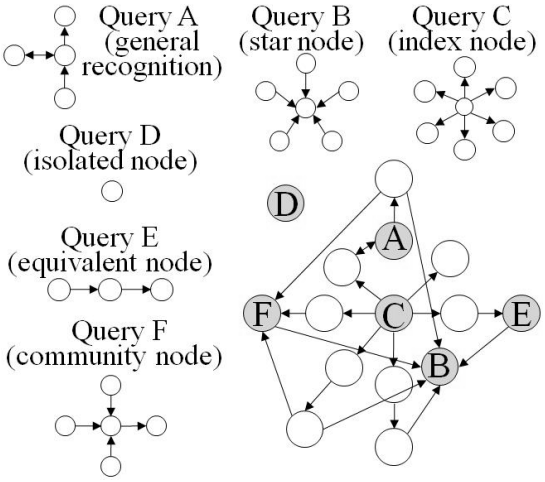

Figure 6. Types of patterns

Query A is an example of general recognition. Query A consists of the number of inlinks, outlinks, pairlinks and total nodes. Then, the system finds the matching patterns against the query. Query B is about a star node. A user asks which node has the maximum number of inlinks. It is the most attractive node in its network. Query C is for an index node; a user asks which node has the maximum number of outlinks. In Web site, it is a index page. In Java API Docs, it is a typical interface. In sociometry, the person is the most popular person. Query D asks an isolated node. A user asks which node has no inlink, outlink and pairlink. Then, the system finds the nodes that are isolated in their network. Query E asks an equivalent node. A user asks which nodes have the same number of inlinks and outlinks. Query F is a community node. A user asks which nodes have the number of inlinks which is 10% more than that of outlinks.

4. The Structure of the System

Figure 7 depicts the architecture of the proposed system. Since it requires computationally expensive works, we use

JXTA technologies [15] that allow developers to implement distributed services and applications. Even though we use a master peer for the preprocessing and recognition task, it can be extended to perform computational intensive works by using slave peers.

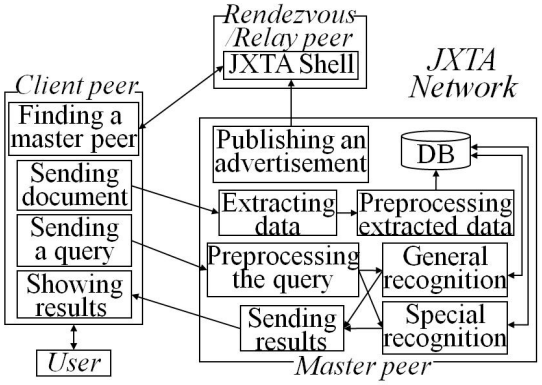

Figure 7. The architecture of our system

A rendezvous peer manages peer's pipe advertisements. A master peer publishes its pipe advertisement to the rendezvous peer. A client peer joins in the JXTA network and finds the mater peer which preprocesses and recognizes matching patterns against a query or special nodes.

For preprocessing, the client peer sends documents to be preprocessed such as Web pages, OWL, RDF, or Java Docs to the master peer. The master peer extracts data from the documents such as in-link, out-link, and pair-link. If the documents are OWL or RDF documents, it uses Protégé API [9]. If they are Web pages or Java Docs, it uses HTML Parser. The master peer preprocesses the extracted data by constructing geometric patterns, computing their invariant properties, building a hash table and saving the information related to each pattern in a database.

For recognition, the user can send two types of queries. A general query is to find a certain node and a special query is to find unique types of nodes such as star, index, isolated, equivalent, and community node. The master peer searches matching patterns or the unique nodes and sends the results to the client peer. The client peer shows them to the user.

5. Illustrative Examples

In this section, we show how various types of patterns can be found.

5.1 General recognition

This example shows how the patterns in figure 1 in section 1 can be found.

1. It counts the number of in-links (I), out-links (O) and pair-links (P) for each node in network n.

	A	B	C	D	E	F	G
I	5	1	1	1	1	1	1
O	2	1	2	1	1	2	2
P	2	0	1	1	0	1	1

2. It divides the number of in-links (I), out-links (O) and pair-links by N-1, where N is the total number of nodes in the network.

	A	B	C	D	E	F	G
I.S	0.833	0.167	0.167	0.167	0.167	0.167	0.167
O.S	0.333	0.167	0.333	0.167	0.167	0.333	0.333
P.S	0.333	0	0.167	0.167	0	0.167	0.167

3. It represents a node as a pattern based on I.S(x), O.S(y) and P.S (z) by 3D coordinates by connecting each pair of two values and making a segment.

4. It calculates the relative position of the node. (I.O.S = I.S - O.S)

	A	B	C	D	E	F	G
I.O.S	0.5	0	-0.167	0	0	-0.167	-0.167

5. It calculates cosines of each angle of the pattern (cos α_{pi}, cosα_{pj}, cos α_{pk})

	A	B	C	D	E	F	G
cos α_{pi}	0.263	0.707	0.8	0.5	0.707	0.8	0.8
cos α_{pj}	0.263	0	0.316	0.5	0	0.316	0.316
cos α_{pk}	0.862	0.707	0.316	0.5	0.707	0.316	0.316

6. It calculates a hash value h(k_p) of the pattern with the invariant properties (cos α_{pi}, cosα_{pj}, cos α_{pk}, I.O.S_p).

	A	B	C	D	E	F	G
h(k_p)	2754	3095	2143	558	3095	2143	2143

7. It inserts a slot h(k_p) in a hash table T. An element in the hash table T is a hash value which is connected to the invariant properties of the pattern which are stored in a database.

The query pattern in figure 1 has one in-link, two out-links and one pair-link. The steps for recognition are as follows.

1. It constructs a pattern of a query node.

I.S	O.S	P.S	I.O.S	cosα_{pi}	cosα_{pj}	cosα_{pk}
0.167	0.333	0.167	-0.167	0.8	0.316	0.316

2. It calculates a hash value h(k_p) of the query pattern. (h(k_p) = 2143)

3. It searches candidate patterns of which the hash values are equal to that of the query pattern. (candidate patterns : node C, node F, and node G)

4. It compares cosines of the query node with those of the candidate patterns.

5. It votes matching patterns if the gap between their cosines and the cosines of the query pattern is below a threshold t. (matching patterns : node C, node F, and node G)

Figure 8 is the screenshot of the system that a user can see for general recognition. A user chooses a tab "General Recognition" and type values of in-link, out-link, and pair-link for a target node in a network. Then, the user clicks a button "send" after typing a query. It finds three patterns (i.e., c, f, g).

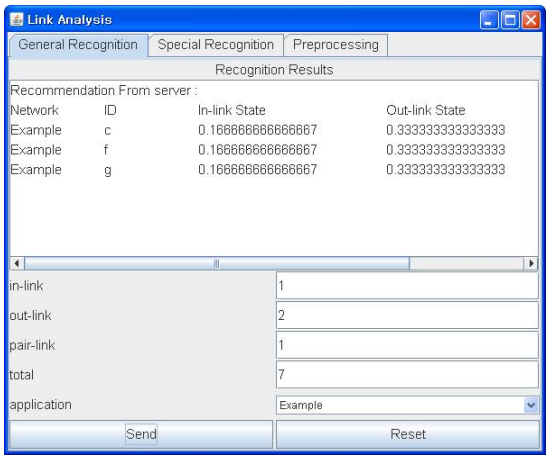

Figure 8. General recognition

5.2 Special recognition

In this example, we show how different types of patterns can be found using our system. Figure 9 shows the network from which patterns are found.

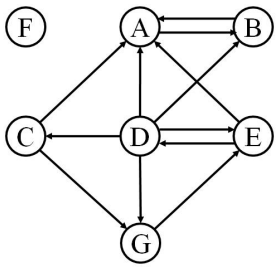

Figure 9. An example network

Preprocessing is done as follows.

1. It counts the number of in-links (I), out-links (O) and pair-links (P) per each node.

	A	B	C	D	E	F	G
I	4	2	1	1	2	0	2
O	1	1	2	5	2	0	1
P	1	1	0	1	1	0	0

2. It divides the number of in-links (I), out-links (O) and pair-links by N-1, where N is the total number of nodes in the network. (I.S = I/ N−1, O.S = O / N−1, P.S = P /N)

	A	B	C	D	E	F	G
I.S	0.667	0.333	0.167	0.167	0.333	0	0.333
O.S	0.167	0.167	0.333	0.833	0.333	0	0.167
P.S	0.167	0.167	0	0.167	0.167	0	0

3. It represents a node as a pattern based on I.S(x), O.S(y) and P.S (z) by 3D coordinates by connecting each pair of two values and making a segment.

4. It calculates the relative position of the node. (I.O.S = I.S - O.S)

	A	B	C	D	E	F	G
I.O.S	0.5	0.167	-0.167	-0.667	0	0	0.167

5. It calculates cosines of each angle of the pattern (cos α_{pi}, cosα_{pj}, cos α_{pk})

	A	B	C	D	E	F	G
cos α_{pi}	0.171	0.316	0.894	0.962	0.632	0	0.447
cos α_{pj}	0.171	0.316	0	0.139	0.2	0	0
cos α_{pk}	0.941	0.8	0.447	0.139	0.632	0	0.894

6. It calculates a hash value h(k_p) of the pattern with the invariant properties (cos α_{pi}, cosα_{pj}, cos α_{pk}, I.O.S_p).

	A	B	C	D	E	F	G
h(k_p)	4833	588	256	3131	3218	0	3739

7. It inserting a slot h(k_p) in a hash table T. An element in the hash table T is a hash value which is connected to the invariant properties of the pattern which are stored in a database.

There are five types of patterns that special recognition considers in our system. They are index node, star node, isolated node, equivalent node, and community node.

1. Index node - the index node has the least state value (I.O.S). The rate of the outlink is much more than the rate of the inlink. In this

network, the index node is node D whose state value is -0.667.

Figure 10 is the screenshot of the system that a user can see for index node recognition that is one of six possible types of special recognition. A user selects a tab ”Special Recognition” and chooses a type of a unique node. There are six types of nodes (i.e., index node, star node, isolated node, equivalent node, high node, and community node). There are the searching results when a unique node type is selected. A value ”state” decides the unique node type (state = inlink/(total-1)-outlink/(total-1)). The index node has the least state value. The rate of the outlink is much more than the rate of the inlink.

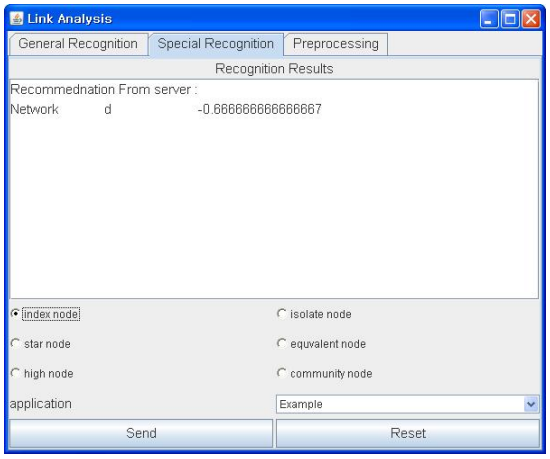

Figure 10. Index node recognition

2. Star node - the star node has the most state value. The rate of the inlink is much more than the rate of the outlink. In this network, the star node is node A whose state value is 0.5.

3. Isolated node - the isolated node has neither inlink nor outlink. It is isolated from the network. In this network, the isolated node is node F that has no inlink and outlink.

4. Equivalent node - the inlink value of an equivalent node is equal to its outlink value. In this network, the equivalent node is node E that has two inlinks and two outlinks.

5. Community node - the state value of the community node is more than 0.1. It means that the rate of inlink is more than 10 percent the rate of outlink. In this network, the community nodes are node A (I.S.O=0.5), B (I.S.O=0.167), and G (I.S.O=0.167).

6. Experimental Results

For the experimentation, we check whether our system finds all real nodes that match the pattern in the network using recall and precision, where recall is the ratio between matching nodes found and all true nodes in the network and precision is the ratio between real nodes and retrieval results.

First, we experiment the system with the blog data. We choose blogs in a portal site called ’NAVER’ [5]. There are two types of links that are ’blog neighbor’ and ’neighbor blog’. The blog neighbor means an in-link which is referred by other blogger. The neighbor blog means an outlink which refers to other blogger. We calculate the number of each of them.

We preprocess data about link relationships among web communities. We choose 30 community pages that are blogs. It computes cosines of each angle of a pattern that are invariant properties ($\cos\alpha_{pi}$, $\cos\alpha_{pi}$, $\cos\alpha_{pk}$). It computes also I.O.S that is a position in a network based on in-links and out-links from other communities. The three cosines and a position ($\cos\alpha_{pi}$, $\cos\alpha_{pi}$, $\cos\alpha_{pk}$, I.O.S_p) are used to generate slot indices for a hash table. Each slot corresponds to a record of fields in a database where we store information of each pattern such as cosines, I.O.S, community id and network id.

Data recognition is done as follows. It computes the invariant properties of a query. Then, it searches indices of candidate patterns that have the hash value equal to the query. It compares the cosines between the query and candidate patterns. It casts a vote to the patterns below the threshold after checking whether each difference between cosines is below the threshold or not. It determines a matching pattern that has the most votes.

The data used for this experiment is as follows.

No	1	2	3	4	5	6
I	100	91	76	62	50	34
O	17	4	2	40	13	3
p	0	0	0	0	0	0
No	7	8	9	10	11	12
I	14	12	11	8	5	4
O	50	5	3	0	20	0
p	0	0	0	0	0	0

No	13	14	15	16	17	18
I	4	1	0	9016	4458	2421
O	0	1	0	716	697	821
p	0	0	0	0	0	0
No	19	20	21	22	23	24
I	2419	1906	1609	1031	642	503
O	0	193	183	114	0	5
p	0	0	0	0	0	0
No	25	26	27	28	29	30
I	365	334	306	260	110	100
O	166	0	164	1	22	206
p	0	0	0	0	0	0

Figure 11 shows the results.

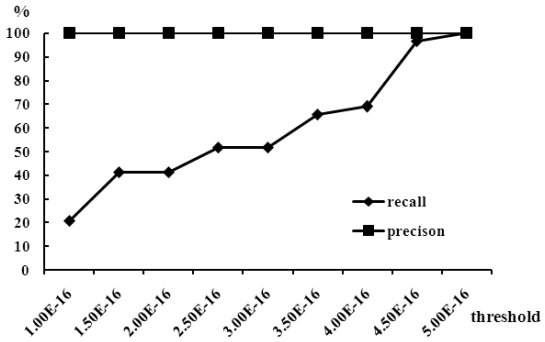

Figure 11. Blog result

Second, we analyze a part of Java tutorial [6]. We exclude all the references that are not local in the web site. We also exclude all the references with the same destination. We measure the number of incoming and outgoing links of each web page from other web pages. There are 31 web pages in a web site. The data used for this experiment is as follows.

No	1	2	3	4	5	6	7	8
I	1	2	2	2	2	2	2	4
O	6	2	2	2	2	2	2	2
p	1	2	2	2	2	2	2	2
No	9	10	11	12	13	14	15	16
I	2	2	3	2	2	2	2	2
O	2	2	2	2	3	2	2	2
p	2	2	2	2	2	2	2	2
No	17	18	19	20	21	22	23	24
I	5	1	3	1	3	2	2	3
O	4	2	3	1	2	2	2	2
p	4	1	3	1	2	2	2	2
No	25	26	27	28	29	30	31	
I	1	3	5	1	3	2	1	
O	1	2	4	2	2	2	1	
p	1	2	4	1	2	2	1	

Figure 12 shows the experimental result for the web pages data.

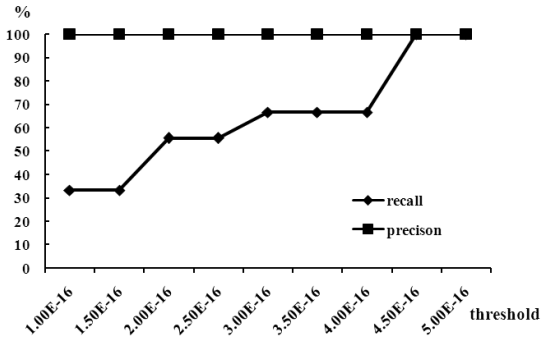

Figure 12. Web page result

Third, we experiment the system with classes in an OWL document that defines travel ontology and there are 35 classes. There are terms in OWL but we use a feature related to RDF Schema (rdfs:subClassOf) and a feature related to equality (equivalentClass) because they are applied to class expressions. The data used for this experiment is as follows.

No	1	2	3	4	5	6	7
I	3	0	0	0	1	0	0
O	0	1	1	1	1	1	0
p	1	0	1	0	0	0	0
No	8	9	10	11	12	13	14
I	4	3	0	0	2	0	0
O	0	1	1	2	1	1	1
p	0	0	0	0	0	0	0
No	15	16	17	18	19	20	21
I	3	0	0	2	0	0	0
O	1	1	1	1	1	1	0
p	0	0	0	0	0	0	0
No	22	23	24	25	26	27	28
I	3	0	0	0	0	0	0
O	0	0	1	0	0	0	0
p	5	1	0	1	1	1	1
No	29	30	31	32	33	34	35
I	2	0	0	2	1	0	0
O	1	1	1	1	1	1	1
p	0	0	0	0	0	0	0

Figure 13 shows the experimental result for the OWL documents.

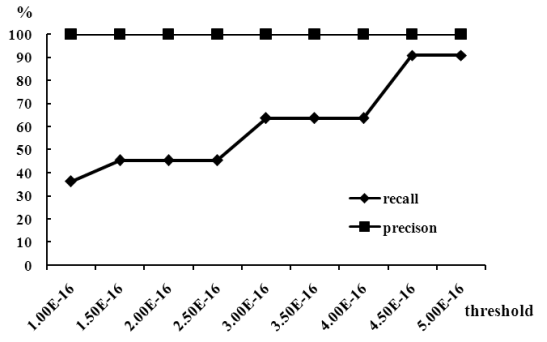

Figure 13. OWL result

Fourth, we use our system for measuring relationships between students in a classroom like sociometry. All of the students are 20. The data used for this experiment is as follows.

No	1	2	3	4	5	6	7
I	1	4	3	2	4	12	7
O	0	0	0	2	4	0	0
p	0	3	3	1	1	4	4
No	8	9	10	11	12	13	14
I	7	4	7	8	12	6	7
O	2	0	0	0	1	1	1
p	2	3	1	4	5	3	4
No	15	16	17	18	19	20	
I	0	2	3	7	0	4	
O	0	0	0	0	4	0	
p	0	2	2	5	0	3	

We give them questions related to positive and negative nomination and collect their answers. We measure their relationships by using the nomination which each student receives from their peers. We regard the positive nomination as the inlink, the negative nomination as the outlink, and the mutual nomination as pairlink. We can make a network of the friendships. Figure 14 presents the experimental result for students' friendships.

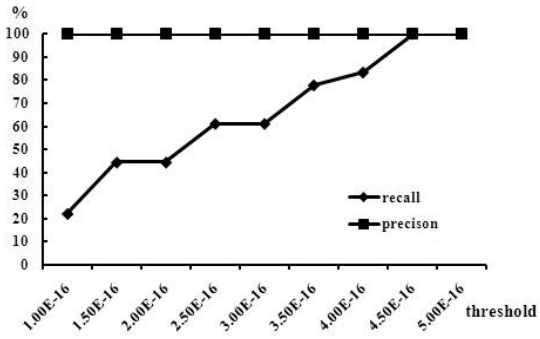

Figure 14. Sociometry result

7. Conclusions

Pattern recognition plays an important role in various application areas [16]. In this paper, we present a system that can be used to find patterns in a social network using geometric hashing. We experimented the system against four types of social network; blog data, web pages, OWL documents, and socio-metric data. In all cases, the results indicated that the proposed system can find various types of patterns in a social network correctly.

Our system can find high or low ranked page so that we can use the information to restructure the network. For example, our system can find isolated nodes that are not linked from any page and do not have links to any page. If its content is necessary to describe a certain topic, we can create new links from the page to other related pages. We can also detect the degree of coupling of the network. If there are many isolated nodes, the network is the low degree of coupling and so we can reconstruct its link structure to improve its degree of coupling. Our system can find index nodes that are not linked from any page but have many links to other pages. We can use it as a place to announce or advertise.

The proposed system can help measure the navigability of a certain network by analyzing in-links, out-links, the number of nodes in the network. The more complex that the web pages are interlinked, the more likely that the users become lost in the hyperspace, and so the harder to navigate [8]. Therefore it is important to design a well designed web site that reduces the symptom of getting lost [17]. We can know the distribution of the links within a website by using our system because it can search isolated nodes in a network. Therefore, our system can be used to measure structural complexity of a website.

A node role is used to explain the behavior of a node in relationship to its neighbors and to the network at large. Knowing the role that a node plays is important for link mining applications [11]. Our system can recognize nodes with unique features within a network by using its link structure. We identify 5 basic types of nodes according to the features such as star, index, isolated, equivalent and community node. These types of nodes can be suited to certain roles. For example, a blog is a star node which has the most incoming links in a community. The node might be well known to the members. If someone wants to notice special information quickly, putting the information on the star node can be the best way to spread it. A page is an index node that has the most outgoing links in a web site. The node might point to many other pages directly. If a user accesses to the web site for the first time, the user can easily reach many nodes by starting from the index node. It can improve novices' accessibility to the web site.

REFERENCES

1. GOTTA, M., O'KELLY, P., **Trends in Social Software**, Educause Center for Applied Research, 1993.
2. KLEINBERG, J. M., **Authoritative Sources in a Hyperlinked Environment**, Journal of the ACM, 1999, Vol. 46, No. 5, pp. 604–632.
3. LIU, B., **Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data**, Data-Centric Systems and Applications, 2007.
4. NAKAKUBO, H., S. NAKAJIMA, K. HATANO, J. MIYAZAKI, S. UEMURA, **Web Page Scoring Based on Link Analysis of Web Page Sets**, Proceedings of the 18th International Workshop on Database and Expert Systems Applications, 2007, pp. 269–273.
5. **NHN Crop**, http://section.blog.naver.com, 2009.
6. **Sun Microsystems Inc.**, http://java.sun.com/docs/books/tutorial/java/javaoo/index.html, 2008.
7. WOLFSON, H. J., **Geometric Hashing: An Overview**, IEEE Computational Science & Engineering, 1997, Vol. 4, pp. 10–21.
8. ZHANG, Y., H. ZHU, S. GREENWOOD, **Web Site Complexity Metrics for Measuring Navigability**, Proceedings of the 4th International Conference on Quality Software, 2004, pp. 172–179.
9. **Protégé-OWL API**, http://protege.stanford.edu/plugins/owl/api, Stanford Center for Biomedical Informatics Research, 2010.
10. **OWL**, http://www.w3.org/2004/OWL, W3C, 2007.
11. SCRIPPS, J., P. N. TAN, A. H. ESFAHANIAN, **Node Roles and Community Structure in Networks**, Proceedings of the 9th WEBKDD and 1st SNA-KDD Workshop '07, ACM, 2007, pp. 26-35.
12. JUNG, H., AHN, J., S. PARK, **A JXTA-based System for Protein Structure Comparison**, The Journal of Korean Association of Computer Education, Vol. 12, No. 4, 2009, pp. 57–64.
13. JUNG, H., S. PARK, **A JXTA-based Music Information Retrieval System**, Proceedings of the 5th International Conference on Information Systems Technology and its Applications, Vol. 84, 2006, pp. 107–117.
14. ARASU, A., J. NOVAK, A. TOMKINS, J. TOMLIN, **PageRank Computation and the Structure of the Web: Experiments and Algorithms**, Proceedings of the Eleventh International World Wide Web Conference, Poster Track, 2002, pp. 107–117.
15. GONG, L., **Project JXTA: A Technology Overview**, Sun Microsystems, Inc., 2001.
16. WACHS, J. P., **Gaze, Posture and Gesture Recognition to Minimize Focus Shifts for Intelligent Operating Rooms in a Collaborative Support System**, International Journal of Computers, Communications & Control, Vol. V, Number 1, 2010, pp. 106-124.
17. BANCIU, D., **e-Romania – A Citizens' Gateway towards Public Information**, Studies In Informatics and Control, Vol. 18, Number 3, 2009, pp. 205-210.