Finding Meaning of Clusters

Yutaka Matsuo
National Institute of Advanced Industrial
Science and Technology (AIST)
Aomi 2-41-6, Tokyo 135-0064, Japan
y.matsuo@aist.go.jp

Yukio Ohsawa
Japan Science and Technology Corporation
Tsutsujigaoka 2-2-11
Sendai 983-0852, Japan
osawa@gssm.otsuka.tsukuba.ac.jp

Abstract

Clustering is an important data exploration task in chance discovery as well as in data mining. The first hierarchical clustering dates back to 1951 by K. Florek; since then, there have been numerous algorithms. However, there is no consensus among researchers as to what constitutes a cluster; the choice of the cluster is application-dependent. Although clustering is sometimes evaluated by interpretability of clusters, few studies have been done to reveal the interpretation aspect of clusters. This paper explains development of a new clustering algorithm by graph-based partitioning which aims to simplify interpretation of clusters. Two typical cluster types are considered: a star and a diamond. A star is a cluster with explicit shared context, represented by a central node. A diamond is a cluster with shared context, whose main cause of the context is implicit and hidden. These two types are very easy to understand. We elicit these types of clusters from a given weighted linkage graph. Minimization of weight of the graph cut is also considered. We show some examples and explain the effectiveness of our method.

Introduction

In the 1960s, Stanley Milgram showed that two randomly chosen individuals in the United States are linked by a chain of six or fewer first-name acquaintances, known as "six degrees of separation"[1]. Watts and Strogatz defined the small world mathematically and showed that some networks have small-world characteristics (Watts & Strogatz 1998). Since their introduction, small-world networks and their properties have received considerable attention. Numbers of networks are shown to have a small-world topology. Examples include: social networks, such as acquaintance networks and collaboration networks; technological networks, such as the Internet, the World-Wide Web, and power grids; and biological networks, such as neural networks, foodwebs, and metabolic networks (For reference, see (Girvan & Newman 2002)). Matsuo et al. showed that word co-occurrence in a technical paper also produces a small-world graph (Matsuo, Ohsawa, & Ishizuka 2001b).

[1]Afterwards, much discussion began to address the interpretation of this experiment.

In a small-world graph, nodes are highly clustered, yet the path length between them is small. Although some recent works have proposed different definitions of "small world" (e.g., (Marchiori & Latora 2000)), one by Watts and Strogatz appropriately grasped ideas of node distance and clusters. They define the following two invariants (Watts & Strogatz 1998):

- *Characteristic path length* L is the path length averaged over all pairs of nodes. Path length $d(i, j)$ is the number of edges in the shortest path between nodes i and j.
- *Clustering coefficient* is a measure of cliqueness of local neighborhoods. For a node with k neighbors, at most, ${}_kC_2 = k(k-1)/2$ edges can exist between them. Clustering of a node is the fraction of these allowable edges that occur. Clustering coefficient C representes average clustering over all nodes in the graph.

Watts and Strogatz define a small-world graph as one in which $L \geq L_{rand}$ (or $L \approx L_{rand}$) and $C \gg C_{rand}$, where L_{rand} and C_{rand} are the characteristic path length and clustering coefficient of a random graph with the same number of nodes and edges.

This paper proposes a new method for detecting clusters based on clustering coefficient C. In a small-world network, shortcuts (or weak ties) play an important role in connecting clusters (or communities). In other words, clusters already exist in the graph. Therefore, we eliminate some edges from the graph to make C large so that nodes are clustered and clusters are separated.

Finding clusters (or rather, "eliciting" of clusters when they exist in nature) is an important task when we try to understand the graph. A cluster often shows the particular context; for example, a cluster corresponds to a community in social networks, a customer group, in a Web page community on the WWW, or a concept in a technical paper. Proper clustering may suggest the context shared by the member of each cluster, as well as facilitate preprocessing of data before statistical analysis .

Related Works

Clustering is a division of data into groups of similar objects. Each group, called a cluster, consists of objects that are similar among themselves and dissimilar to objects of

other groups. Representing data by fewer clusters necessarily ignores certain fine details, but achieves simplification (Berkhin 2002). Clustering is an important data exploration task in chance discovery (Ohsawa 2002) as well as in data mining. It shows an overview of the data, elucidates the data, and stimulates new ideas.

Clustering techniques are generally divided in hierarchical clustering and partitioning. The first hierarchical clustering dates back to 1951 by K. Florek; since then, there have been numerous modifications. One of the most widely-used clustering methods is single linkage clustering. It is a kind of hierarchical clustering; its cluster relationships can be represented by a rooted tree, called a dendrogram. A cluster is produced by cutting edges of the dendrogram with a threshold. However, application of only one threshold for all clusters would produce many small clusters and a few large clusters.

To tackle this problem, a clustering method based on a linkage graph is proposed in Kawaji *et al.* (2001) to cluster protein sequences into families. They formulate clustering as a kind of graph-partitioning problem (Fjällström 1998) of a weighted linkage graph and find a *minimal cut* with consideration of balancing cluster size. The graph partitioning problem is of interest in areas such as VLSI placement and routing, and efficient parallel implementations of finite element methods, e.g., to balance the computational load and reduce communication time. Flake, Lawrence, & Giles (2000) developed an algorithm to find communities on the Web by maximum-flow / minimum-cut framework. This algorithm performs well in practice, i.e., under the condition that one doesn't have rapid access to the entire Web.

Another approach focuses on the *betweenness* of an edge in a linkage graph as the number of shortest paths between pairs of nodes that run along it (Girvan & Newman 2002). Thereafter, we call this method *betweeness clustering*. Edges in the graph are iteratively removed if the betweenness of the edge is highest; this reveals communities in social and biological networks.

Han *et al.* (1998) used association rules and hypergraph machineries. First, frequent item-sets are generated from transactional data. A hyper-graph $H = (V, E)$ can be associated with an item universe such that vertices V are items. In a common graph, pairs of vertices are connected by edges, but in a hypergraph, several vertices are connected by hyperedges. Then, a solution to the problem of k-way partitioning of a hyper-graph H is obtained.

A series of chance discovery studies have targeted mining from natural language documents and transactional data. Transactional data relates to clustering of categorical data. Every transaction can be presented in a point-by-attribute format, by enumerating all items j, and by associating a transaction with the binary attributes that indicate whether j-items belong to a transaction or not. Such data often have high deminsionality, a significant amount of zero values, and a small number of common values between two objects. Conventional clustering methods, based on similarity measures, do not work well. Since categorical/transactional data are important in customer profiling, assortment plannning, Web analysis, and other applications, a different clustering method founded on the idea of *co-occurrence* of categorical data has been developed (Berkhin 2002). CACTUS (Ganti, Gehrke, & Ramakrishnan 1999) seeks hyper-rectangular clusters in point-by-attribute data with categorical attributes, based on co-occurrence of attribute-value pairs. Two values a and b of two different attributes are strongly connected if the number of data points having both a and b is larger than the frequency expected under an independency assumption by a user-defined margin.

The clustering system can be assessed by an expert, or by a particular automated procedure. Traditionally, this assessment relates to two issues: (1) cluster interpretability, and (2) cluster visualization. However, little work has been done so far for considering the ease of cluster interpretation. Our reseach aims at improving cluster interpretability.

SD Clustering

Clustering criterion

Recent studies show that many networks have small world characteristics in nature. It means nodes are highly clustered and a few edges play a role as shortcuts between clusters. Therefore, by eliminating those shortcuts, we can elicit clusters.

Small worlds are characterized by large C and small L. In this context, a graph with separated clusters (called a *cave world* by Watts's terminology) has large C and large L. So, we would like to get a graph with large C and large L by eliminating some edges.

However, preliminary experiments show that merely maximizing C will bring good clustering. So, we formalize our clustering algorithm, so-called *SD clustering*, as follows.

Definition 1 (SD Clustering) *Given a graph $G = (V, E)$ and k where V is a set of nodes, E is a set of edges, and k is a positive integer,* SD clustering (SDC) *is defined as finding a graph G' with k clusters by removing edges so that*

$$f = C_{G'} \tag{1}$$

is to be maximized where $C_{G'}$ is C for graph G'.

SDC differs from the conventional graph partition problem (Fjällström 1998) in that it uses C for measurement.

Figure 1 is a sample graph derived from co-occurrence of words in a technical paper. Each node represents a word, and each edge represents co-occurrence of two words. Applying the SD clustering into 10 clusters, we get the clusters shown as Fig. 2. In the graph, many complete subgraphs (or *diamonds*) and hubs (or *stars*) emerge. Figure 3 shows an illustration of two types of subgraphs. A diamond is a subgraph where $C = 1.0$; a star is a subgraph where C is nearly 1.0. ($C = 1$ for each surrounding nodes and $C < 1$ for the central node.) Therefore, these two clusters are likely to emerge when we maximize C. We call our algorithm SD (Star and Diamond) clustering because we can elicit many stars and diamonds from the graph.

We take these two as typical cluster types which are easy to understand and interpret. A star shows shared context around the central node. The main cause of the cluster can

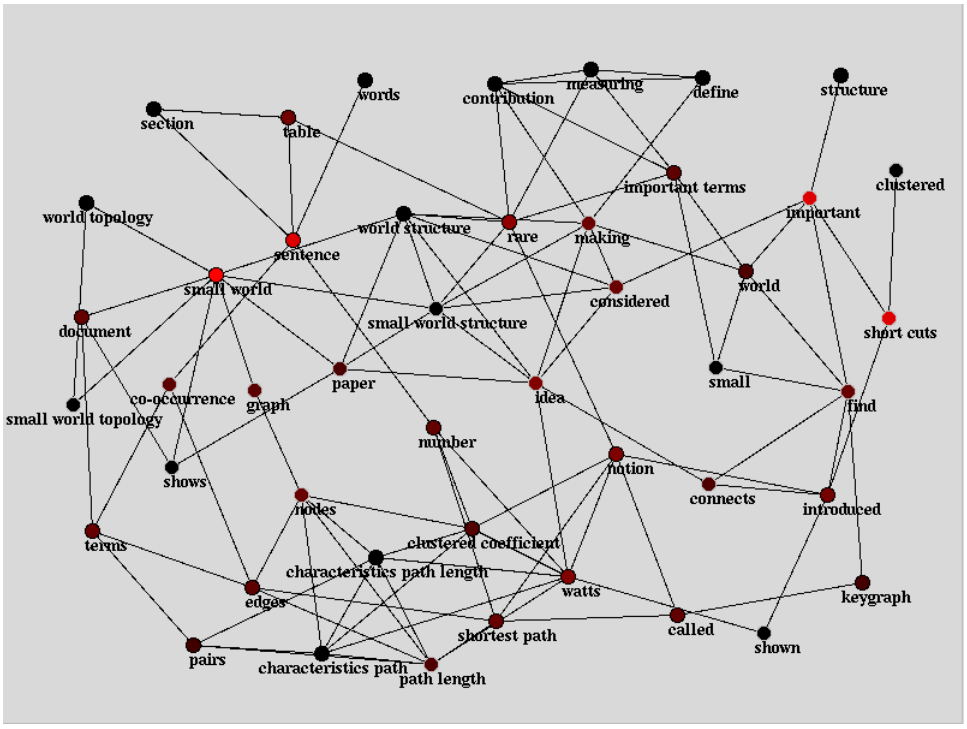

Figure 1: Example of original graph. $C = 0.429$.

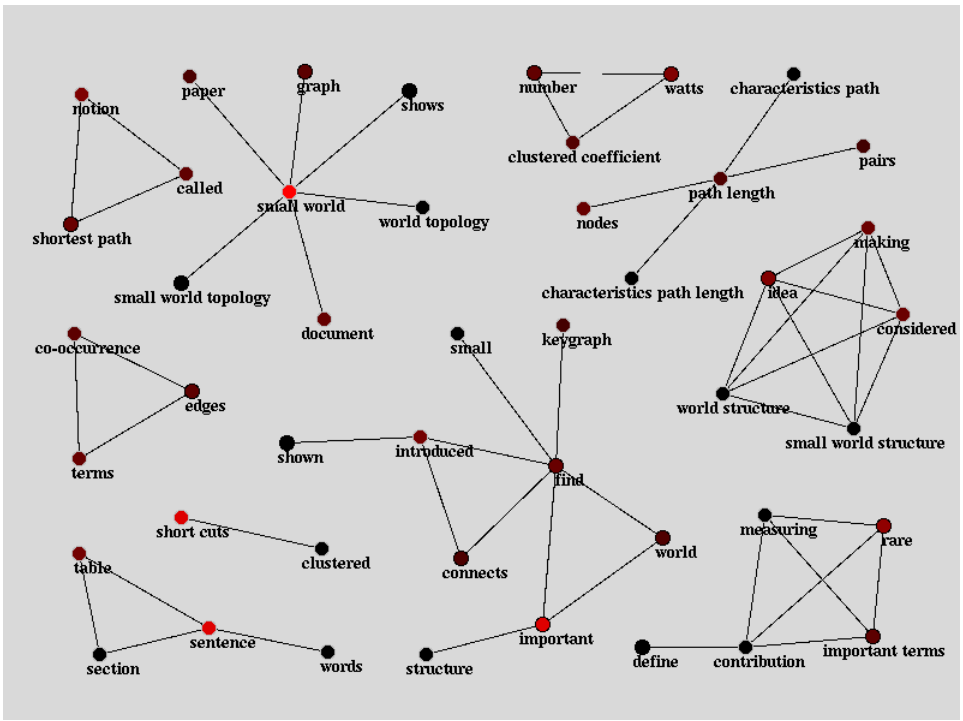

Figure 2: Example of SD clustering. $C = 0.883$.

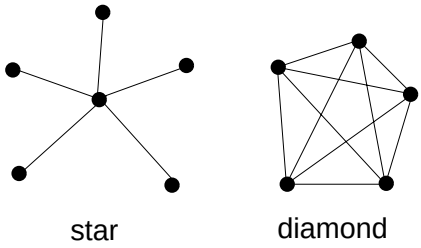

Figure 3: Illustration of a diamond and a star.

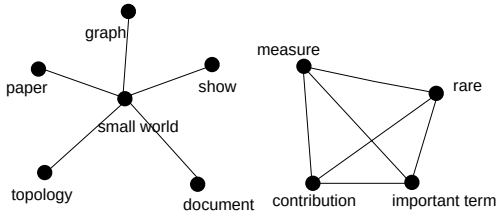

Figure 4: Example of a diamond and a star.

be grabbed by the central node. A diamond shows the shared context of all nodes, where the main cause is hidden behind the cluster. For example, the left graph of Fig. 4 shows a star, which consists of the words all related to "small world" in the technical paper. The cluster can be interpreted as the concept "small world". On the other hand, the right graph of Fig. 4 is a diamond. The words are all related to each other in the same context. Although the context is not apparently represented as a central node of a star, there might be a hidden cause of the cluster. In this case, these words are related to the concept of "keyword" in the paper.

Thus, we think diamonds and stars are a very appropriate form of clusters to ease the interpretation of clusters. However, we have to consider which edge to cut at the same time because, if we are allowed to cut any edges, we can freely make stars or diamonds. The cost to cut the edge is also considered below. SDW (Star and Diamond Weighted) Clustering is an extended version of SD clustering with consideration of edge weight.

Definition 2 (SDW Clustering) *Given a weighted graph $G = (V, E)$ and k where V is a set of nodes, E is a set of edges, and k is a positive integer,* SDW clustering (SDWC) *is defined as finding a graph G' with k clusters by removing edges so that*

$$f = a \times C_{G'} - b \times W_{G'} \quad (2)$$

is to be maximized where $C_{G'}$ is C for graph G', $W_{G'}$ is the sum of the weight of removed edges, and a and b are constants.

Algorithm

The problem of finding an optimal connection among all possible pairs of nodes of a graph has been proven to be NP-complete. Therefore, we consider an approximate algorithm for SWC as follows.

1. Prune an edge which maximizes f iteratively until k edges are pruned.
2. Add an edge which maximizes f. If an edge to be added is the same as the most previously pruned one, terminate.
3. Prune an edge which maximizes f. Go to 2.

The second and third proceedures are optional. If clustering needs to be finished rapidly, these proceedures can be skipped. However, in some cases, they provide a better solution. In our current implementation, we also use tabu search for escaping from local maxima. Detailed discussion of efficient algorithms is our ongoing theme. However, we should note that a clustering criterion and a clustering algorithm are different. This algorithm is a particular implementation of the small-world clustering criterion.

Examples

We show an example of SDWC applied to a word co-occurrence graph. In short, a word co-occurrence graph is constructed as follows:

1. pick up n frequent words as nodes after stemming and discarding stop words;
2. calculate a Jaccard coefficient[2] for each pair of words and add an edge if the coefficient is larger than a given threshold. The weight of an edge is defined by the Jaccard coefficient.

A word co-occurrence graph is shown to have small world characteristics (Matsuo, Ohsawa, & Ishizuka 2001b).

Figure 5 is a word co-occurrence graph derived from a technical paper (Matsuo, Ohsawa, & Ishizuka 2001a) with single-linkage clustering. We can see a big cluster, one little cluster, and eight singleton clusters. A resultant in a big cluster with many isolated nodes is very common when single linkage clustering is applied. It is very difficult to interpret the clusters.

Figure 6 is a result of betweenness clustering (Girvan & Newman 2002). The cluster size are well balanced compared with single-linkage clustering. However, the location of the center and places demanding attention are not obvious because clusters assume many shapes.

SDC results in many stars and clusters. As shown previously, Fig. 2 is the result of SDC (or SDWC where $a = 1$ and $b = 0$). Each cluster is very easy to interpret and the cluster size is well-balanced. (There is only one singleton cluster.) Figure 7 is a result by SDWC where $a = 1.0$ and $b = 0.01$. Because the term W is, in this case, the magnitude of $20 \sim 30$ and C is below 1, we set b to be 0.01. Some stars and diamonds are now merged, forming new clusters. However, we can still interpret clusters by first understanding each star and diamond component, then considering the meaning of the whole cluster.

Lastly, Fig. 8 is clustering by SDWC where $a = 0$ and $b = 1.0$. It means only the minimization of weight of cut is considered. This is an equivalent algorithm to minimum-cut

[2]The Jaccard coefficient is the number of sentences that contain both words divided by the number of sentences that contain either word.

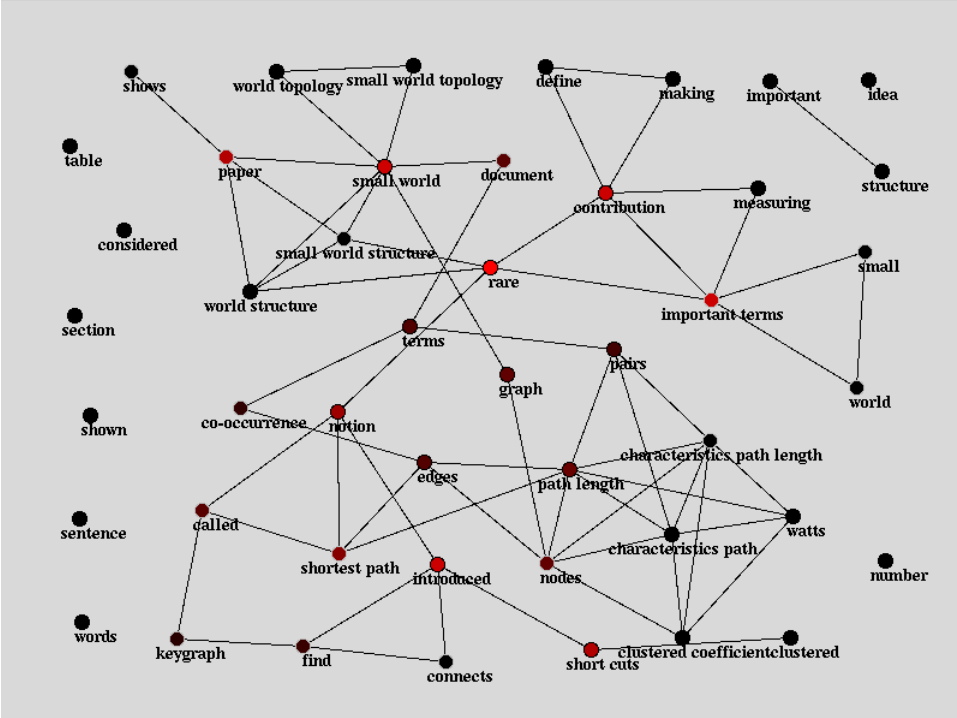

Figure 5: Example of single linkage clustering. $C = 0.452$.

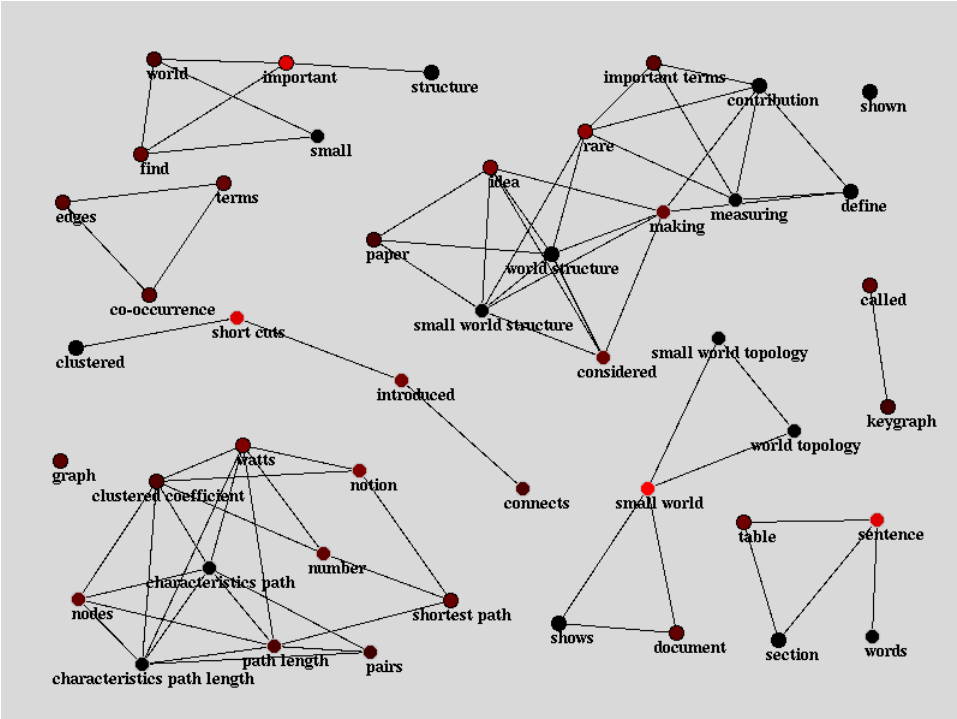

Figure 6: Example of betweenness clustering. $C = 0.679$

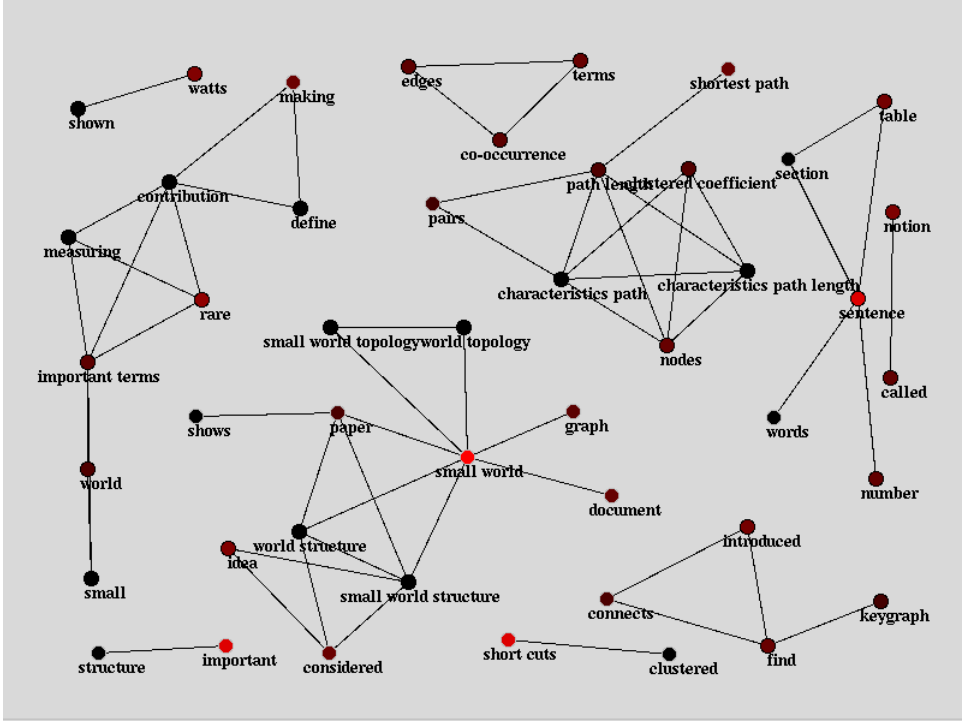

Figure 7: Example of SDW clustering ($a = 1$ and $b = 0.01$). $C = 0.858$.

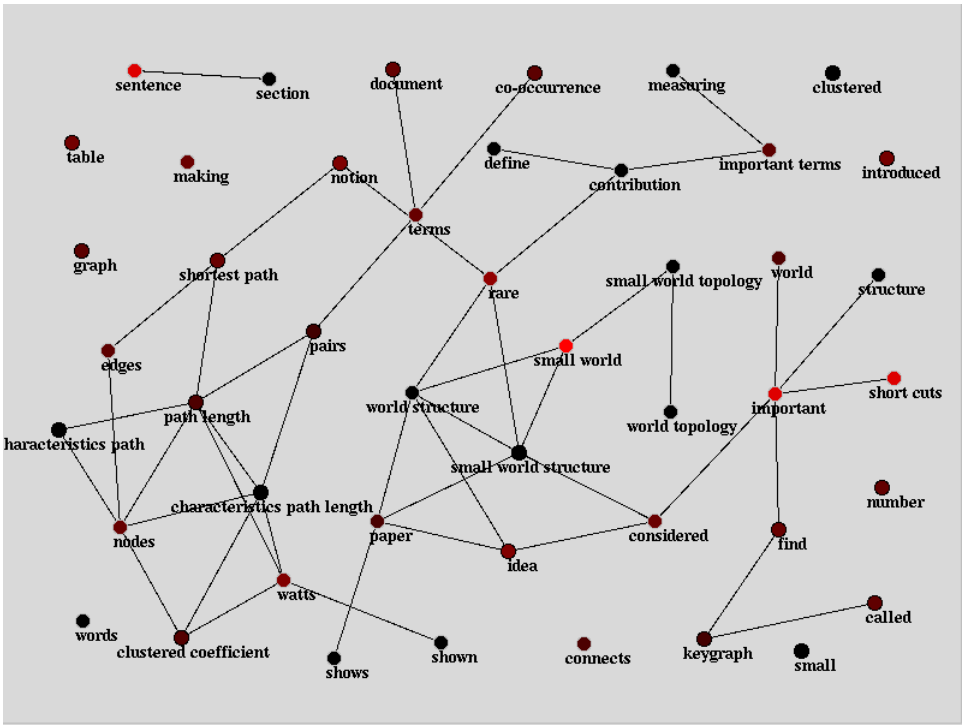

Figure 8: Example of SDW clustering ($a = 0$ and $b = 1.0$). $C = 0.397$.

graph partitioning without a balancing mechanism of cluster size. The obtained clusters are big clusters and many singleton clusters. This result is not good with respect to both cluster size and ease of interpretation. The merit of this algorithm is that it cuts the most trivial part (i.e., the minimal weight edges) of the graph; thus, it represents the data itself well.

Discussion and Conclusion

Clustering is a discovery process that groups a set of data such that intracluster similarity is maximized and the intercluster similarity is minimized (Han *et al.* 1998). C is related to intracluster similarity; if intracluster similarity is large, C is large, and vice versa. Interestingly, maximizing C yields clusters of well-balanced size.

A merit of our algorithm is that the clusters are easy to understand because it usually generate stars and diamonds. In the analysis of Web page in- and out-links, a strongly connected subgraph is often considered as a community (Flake, Lawrence, & Giles 2000). In this sense, what we call a diamon is the same as a community in Web link structure.

Mathias et al. reveals what drives emergence of a small-world network(Mathias & Gopal 2001). They introduce the idea of physical distance between a pair of nodes on top of path length between a pair of nodes; they found an alternate route to small world behaviour through formation of hubs. This is the same as a star, an important component of small-world structure.

From a chance discovery point of view, if a user can find the meaning of clusters easily, she is likely to understand the data well, and gain insight into new perspectives and new ideas. This algorithm can be applied also to the preprocess phase of data mining with human-computer interactions (such as KeyGraph (Ohsawa, Benson, & Yachida 1998)). Quantitative evaluation of interpretability is our future work. We will further investigate what type of clustering can ease the interpretability and stimulate users' imagination.

References

Berkhin, P. 2002. Survey of clustering data mining techniques. http://citeseer.nj.nec.com/526611.html.

Fjällström, P.-O. 1998. Algorithms for graph partitioning: A survey. *Computer and Information Science* 3.

Flake, G. W.; Lawrence, S.; and Giles, C. L. 2000. Efficient identification of Web communities. In *Proc. ACM SIGKDD-2000*, 150–160.

Ganti, V.; Gehrke, J.; and Ramakrishnan, R. 1999. Cactus-clustering categorical data using summaries. In *Proc. 5th ACM SIGKDD*, 73–83.

Girvan, M., and Newman, M. E. J. 2002. Community structure in social and biological networks. *Proceedings of National Academy of Sciences USA* 99:8271–8276.

Han, E.; Karypis, G.; Kumar, V.; and Mobasher, B. 1998. Hypergraph based clustering in high-dimensional data sets: A summary of results. In *Bulletin of the Technical Committee on Data Engineering*, volume 21, 15–22.

Kawaji, H.; Yamaguchi, Y.; Matsuda, H.; and Hashimoto, A. 2001. A graph-based clustering method for a large set of sequences using a graph partitioning algorithm. *Genome Informatics* 12:93–102.

Marchiori, M., and Latora, V. 2000. Harmony in the small-world. *Physica A* 285:539–546.

Mathias, N., and Gopal, V. 2001. Small worlds: How and why. *Physical Review E* 63(2).

Matsuo, Y.; Ohsawa, Y.; and Ishizuka, M. 2001a. A document as a small world. In *Proceedings the 5th World Multi-Conference on Systemics, Cybenetics and Infomatics (SCI2001)*, volume 8, 410–414.

Matsuo, Y.; Ohsawa, Y.; and Ishizuka, M. 2001b. KeyWorld: Extracting keywords from a document as a small world. In *Proceedings the Fourth International Conference on Discovery Science (DS-2001)*.

Ohsawa, Y.; Benson, N. E.; and Yachida, M. 1998. KeyGraph: Automatic indexing by co-occurrence graph based on building construction metaphor. In *Proc. Advanced Digital Library Conference (IEEE ADL'98)*.

Ohsawa, Y. 2002. Chance discoveries for making decisions in complex real world. *New Generation Computing* 20(2).

Watts, D., and Strogatz, S. 1998. Collective dynamics of small-world networks. *Nature* 393:440–442.