Advances in Smart Systems Research : ISSN 2050-8662 : http://nimbusvault.net/publications/koala/assr/
Vol. 3. No. 2 : pp.19-24 : iswk13-005

Analyzing PC Operation Logs by Functionality Clustering of Networks

Kazuhiro SUZUKI [a] Hiroshi YASUDA [a] Kilho SHIN [b] Takako HASHIMOTO [c] and Tetuji KUBOYAMA [d,1]

[a] *School of Science and Technology for Future Life, Tokyo Denki University, Japan*
[b] *Graduate School of Applied Informatics, University of Hyogo, Japan*
[c] *Chiba University of Commerce Coordinator, Japan*
[d] *Computer Centre, Gakushuin University, Japan*

Abstract. This paper aims at analyzing user behavior from PC operation logs by the functionality clustering of networks. The functionality clustering method was proposed by Fushimi *et al*. This method employs PageRank algorithm and classifies nodes according to their functions and roles in the network. In this paper, to extract similar tasks as user behaviors from PC operation logs, we improve the conventional functionality clustering method by transforming PageRank convergence patterns into symbolic representations of time series data, and applying edit distance to these representations. We show that the improved method allows us to classify user behaviors according to their context at work.

Keywords. network clustering, log analysis, PageRank

1. Introduction

Our study aims to find better network clustering. Our purpose is to discover the user behaviors hidden behind the network, in particular, to extract the role of the individual user on a network. Network clustering has various applications. For example, hyperlink, biological, social networks and more. In this study, we apply network clustering method analyzing PC Operation Logs.

In general, user behavior is different between *working time* and *rest time* at work. In this study, we represent user behavior in the form of network. Employees in an IT company work at 9:00 to 17:30. we consider that working time is set to from 9:00 to 17:30 except for rest time. Rest time include after-working time. We propose a method to detect the difference of the usage of computer at working time and rest time.

The diversity of relationships such as friendships and web hyperlinks are regarded as network structures. To discover knowledge from large scale networks, there has been attempts to extract the community behind network structures. The mainstream of community extraction methods is a density based clustering such as Newman method [2]. The density-based approach attempts to find sets of nodes that are connected to each other at a high density in the network. In contract, different approach is necessary to classify

[1] E-mail: ori-kes2013-iimss@tk.cc.gakushuin.ac.jp

nodes that play the similar role on the network. For this purpose, recently, a structure-based network clustering has been proposed. The structure-based clustering method focuses on the structural similarity around individual nodes, and classifies nodes with the similar structure into the same cluster.

There are mainly two approaches in the structure-based clustering:

(1) an *explicit approach* explores the structure in the vicinity of each nodes in an explicit way such as SCAN algorithm [8];

(2) an *implicitapproach* by Fushimi *et al.* [4] partitions the nodes into clusters using convergence patterns of nodes in the PageRank algorithm.

In this paper, we employ the implicit approach by Fushimi [4] due to its simplicity and efficiency (thereafter we called it functionality clustering method).

In the functionality clustering method, This method employed the *k*-median algorithm with a greedy algorithm and the cosine similarity to measure distance the convergence patterns of nodes. The cosine similarity is not suitable for the network that we address because the result is greatly affected by the scale of the network. In this paper, we propose a robust approach to the delay and scale by transforming the convergence patterns into symbolic representations. To show the effectiveness of our improvement, we conduct an experiment for a network structure obtained by PC logs (specifically, window transition logs).

2. Related Work

The functionality clustering method classifies the nodes based on the similarity of the convergence patterns at each node of PageRank. PageRank [1] is the algorithm developed as a method of ranking web pages used in Google. PageRank considers web pages important pointed from the important web pages.

The functionality clustering method by Fushimi *et al.* [4] regards the convergence process of the PageRank score of each node as a time series vector. Next, these vectors are clustered by the greedy *k*-median method with cosine similarity between vectors. Typically *k*-median is robuster against outliers than the average *k*-means, and a greedy algorithm is employed for efficiency by sacrificing accuracy to some extent. The functionality clustering method is summarized as follows, where each feature vector for each node is a sequence of PageRank scores in the convergence process of PageRank, and the number of clusters is denoted by K.

1. **Obtaining feature vectors:** A sequence of PageRank scores in the convergence process is computed for each node v in a given network, and denoted by X_v.
2. **Computing similarity:** For any two nodes u and v, the cosine similarity between X_u and X_v is computed, and denoted by $\mathrm{sim}(X_u, X_v)$.
3. **Clustering:** Nodes in the network are clustered by the greedy k-median method with the similarity $\mathrm{sim}(X_u, X_v)$.

In this paper, we use PC operation logs for analyzing user behavior at work in a company. In prior work, Saito *et. al* [7] analyzes the behavior of PC user by using the hidden Markov model and the kernel PCA of graph structures, and creates a probabilistic model of user behavior.

3. Our method

The functionality clustering method focuses on convergence patterns of PageRank scores for nodes. We would like to consider the similarity of the convergence patterns rather than the values of PageRank scores. The clustering method by Fushimi *et al* is a susceptible to the delay of patterns and scales. Therefore, we propose a robuster method.

3.1. Convergence pattern and Damping Factor

In the Pagerank algorithm, there is a scaling parameter called *damping factor*, which denotes the probability to follow the actual edges in the network. The damping factor affects the speed of convergence of the Pagerank algorithm. As the damping factor approaches to 1, the expected value of the repetition increases dramatically. In the original PageRank paper has proposed to set the value to 0.85 as a trade off between the convergence speed and effectiveness. However, we want to focus on the process of convergence. The network structure should be sensitively reflected on the convergence patterns, we set the damping factor to 0.99.

3.2. Symbolic representation of convergence patterns

We employ a new clustering method for the convergence patterns of PageRank scores. The convergence speeds of PageRank scores and the convergence patterns are different for each node. Thus, we need to take account of the delay and scale of patterns. As a similarity measure between the convergence patterns, the functionality clustering method [4] employs a cosine similarity. The cosine similarity is relatively robust against the scale of patterns, while susceptible to the delay of data.

In our method, we employ SAX [5]. SAX is a well-known method for clustering time-series data. SAX standardizes time-series data, and discretizes the standardized data into symbolic representations. However, SAX is assumed that the time series data follow a normal distribution while PageRank convergence curve, in general, does not follow the normal distribution. In this paper, we employ a different discretization as followings (denoted by $\mathrm{SAX_{UDF}}$).

Figure 1 shows an example of the $\mathrm{SAX_{UDF}}$. Let the feature vector representations of a convergence pattern be $X = (x_1, x_2, \ldots, x_t)$. This vector X is transformed into a symbolic representation $S(X) = s_1 s_2 \cdots s_{t-1}$ as follows.

$$s_t = \begin{cases} \mathrm{U} & (x_{t-1} < x_t) \\ \mathrm{D} & (x_{t-1} > x_t) \\ \mathrm{F} & (x_{t-1} = x_t) \end{cases}$$

In $\mathrm{SAX_{UDF}}$, the distance of symbolic representations is measured by Edit Distance.

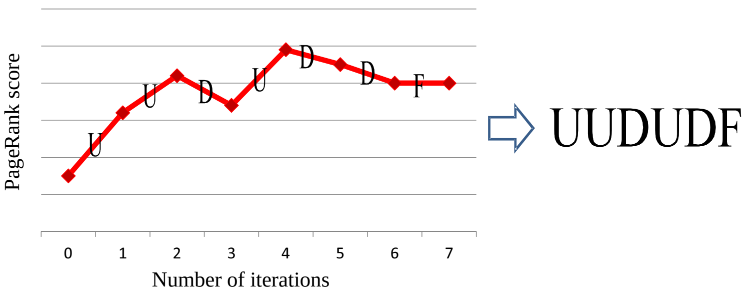

Figure 1. Example of the $\mathrm{SAX_{UDF}}$

4. Experimental Results

We compared the $\mathrm{SAX_{UDF}}$ with the existing functional clustering method. A real network data are used for this evaluation. The network data were generated from PC operation logs of a certain IT company. The company consists of six departments, namely marketing, sales, general affair, management, development and quality assurance departments, and the total number of employees is 63. The company give windows pc each employees and to record the transition of the active window. The PC operation logs includes operation records collected from December to September 2010 that give information of the identifier of PC (PC ID), the user's name, the application name and the time of the event where an active window was switched. Table 1 depicts the log data.

Table 1. PC operational log

PC name	Active app	User name	Time set	Term
09-470	outlook	marketing01	2010/9/1 8:54	2
09-470	excel	marketing01	2010/9/1 8:56	1
09-470	outlook	marketing01	2010/9/1 8:57	1
09-470	word	marketing01	2010/9/1 8:59	2

Figure 2 shows the transitions of active windows: Each node is an active window at some moment, and is attributed with the information of the names of the department and the application and whether the window has become active in the working time. The directed edge between nodes means that the window of the ending node has become active taking over the window of the starting node. The network includes 383 nodes and 4565 edges, and thereafter we call the network *window transition network* for the convenience of explanation.

Also, we can view the network of Figure 2 as a Markov model with transition probability. If a node i in the network has outgoing edges to w_i nodes, the transition probability of each edge is $1/w_i$. Therefore, if v_a of the w_i nodes belong to the same application a, the probability that the node i transits to the application a is v/w_i.

In the comparison, we calculated the distances of the nodes in two different ways: One is based on the functionality clustering method [4], while is based on $\mathrm{SAX_{UDF}}$ that we proposed in this paper. Then, we apply the k-median method to the obtained calculated distances to cluster the nodes.

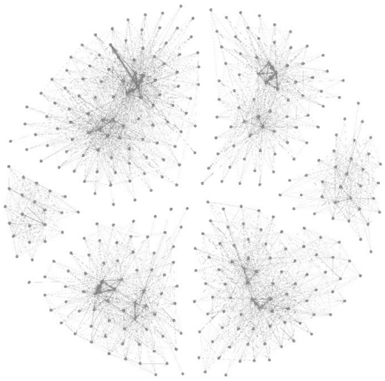

Figure 2. Structure of window transition network

Figure 3 depicts the result of the clustering for the data of the window transition network. For the convenience of explanation, we present the result after reducing the dimensionality by means of Multi-Dimensional Scaling (MDS). Figure 3 shows that our method clearly clusters according to their functionality.

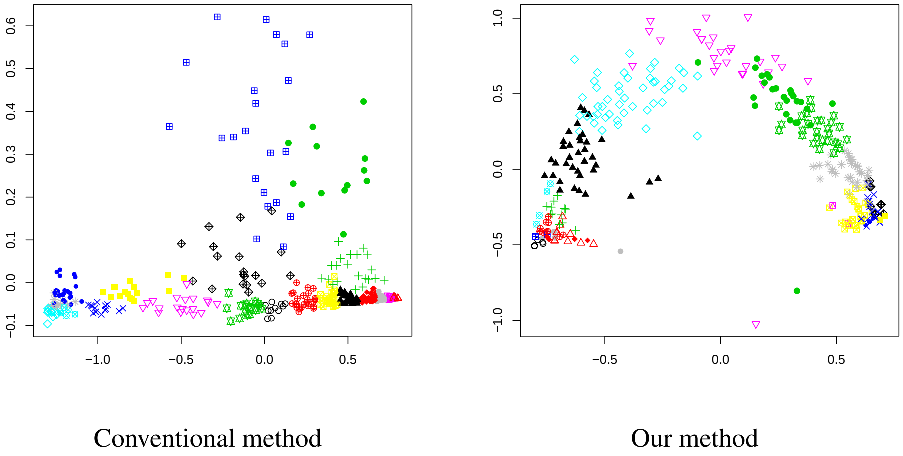

Figure 3. Scatter plots of nodes of window transition network by MDS.

We show the result of clustering for the nodes labeled with Excel in working time or rest time in Table 2. For the transition of active windows, we apply the same k-median method with the cluster number 20, and confirmed whether it would be classified into different clusters with whether or not the working time, even if it was the same application.

The confirmation based on the conditional entropy $H(X \mid Y)$. We let $X \in \{\text{Working time}, \text{rest time}\}$, $Y \in \{\text{cluster1}, \text{cluster2}, \ldots, \text{cluster20}\}$. The conditional entropy $H(X \mid Y)$ for the conventional method is 0.873 while for our method 0.248. This result shows that our

Table 2. Clusters of the nodes labeled with Excel

Department	time	Class	Department	time	Class
marketing	rest time	6	marketing	working time	0
quality assurance	rest time	14	quality assurance	working time	1
management	rest time	6	management	working time	0
sales	rest time	19	sales	working time	0
business office	rest time	19	business office	working time	17
development	rest time	14	development	working time	0

method clearly divides the usage of applications into different cluster according to there time(working time and rest time). This result implies that our method can detect the roles of applications in working time and rest time.

5. Conclusions

In this paper, we aim at analyzing user behavior from PC operation logs by the functionality clustering of networks. The functionality clustering method was proposed by Fushimi *et al*. This method employs PageRank algorithm and classifies nodes according to their functions and roles in the network. In this paper, to extract similar tasks as user behaviors from PC operation logs, we improve the conventional functionality clustering method by transforming PageRank convergence patterns into symbolic representations of time series data, and applying edit distance to these representations. We show that the improved method allows us to classify user behaviors according to their context at work.

References

[1] S. Brin, L. Page: The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 30(1), 107-117 (1998).

[2] A. Clauset, M. E. J. Newman, C. Moore: Finding community structure in very large networks. Phys. Rev. E 70, 066111 (2004).

[3] S. Fortunato: Community detection in graphs.: Physics Reports, 486(3), 75-174 (2010).

[4] T. Fushimi, K. Saito, K. Kazama: Extracting Communities in Networks Based on Functional Properties of Nodes. PKAW 328-334 (2012)

[5] J. Lin, E. Keogh, S. Lonardi, B. Chiu: A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. DMKD '03 Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, 2-11 (2003).

[6] Amy N. Langville, Carl D. Meyer: Google's PageRank and Beyond The Science of Search Engine Rankings. Princeton University (2006)

[7] R. Saito, T. Kuboyama, Y. Yamationkawa, H. Yasuda: Understanding User Behavior through Summarization of Window Transition Logs. LNCS 7108 (DNIS), 162-178 (2011).

[8] X. Xiaowei, Y. Nurcan, F. Zhidan, A. J. S. Thomas: SCAN: a structural clustering algorithm for networks. presented at the Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 824-833 (2007).

[9] Z. Yang, C. Hong, Y. Jeffrey Xu: Graph clustering based on structural/attribute similarities. Proc. VLDB Endow, 2(1), 718-729 (2009).