International Journal of Multimedia and Ubiquitous Engineering
Vol.9, No.4 (2014), pp.13-20
http://dx.doi.org/10.14257/ijmue.2014.9.4.02

Social Network Visualization Method using Inherence Relationship of User Based on Cloud

Yong-Il Kim[1], Yoo-Kang Ji[2] and Sun Park[3*1]

[1]Honam University, South Korea, [2]DongShin University, South Korea, [3]GIST, South Korea

[1]yikim@honam.ac.kr, [2]neobacje@gmail.com, [3]sunpark@nm.gist.ac.kr

Abstract

Most of social network visualization methods have been only focusing on presentation of social network relationship. However, the methods do not consider an efficient processing speed and computational complexity for increasing at the ratio of arithmetical of a big data regarding social network. This paper proposes a cloud based on visualization method to visualize an inherence relationship of user on social network. The proposed method can intuitionally understand the user's social relationship since the method uses correlation matrix to represent a hierarchical relationship of user nodes of social network. It also can easily identify a key role relationship of users on social network. In addition, the method uses Hadoop based on cloud for distributed parallel processing of visualization algorithm, which it can expedite the big data of social network.

Keywords: *Social Network Visualization, SNS, inherence relationship, cloud, Hadoop*

1. Introduction

Every day, we use the digital information with respect to social networks by using file exchanges, chat, blogs, and collective development such as Wikipedia and Open-Source Software projects. All information in today's world can mostly be found online, which has increased the online social networks. According to the increment of online social activities, the necessity of social network analysis (SNA) has been a growing area of the social sciences. The analysis can be automatically instrumented and analyzed for discovering terrorist networks, monitoring outbreaks of diseases, commercial activities, etc. Many of these social networks are large, complex and continuously changing which need an effective approach to help social science researches. Information visualization can well display relationships between users and present their findings to others for SNA [1].

Recently, the approaches of visualizing social networks including node-link (NL) based [2], matrix graph based representations (MAT) [3], and a hybrid based representation with NL and MAT [4, 5] have been proposed. The NL based approaches can be usefully displayed by the overall structure of a network however details about dense sub-graphs is difficult to

[1] Corresponding author

ISSN: 1975-0080 IJMUE

read by the overlapped nodes and the crossed edges [1, 2]. In other words, if the users of social networks are increased, identifying the user's relationship is difficult. In order to resolve the limitation of NL methods, various methods are proposed in connection with post-processing (*i.e.*, sampling, filtering, clustering, etc.) for partial visualization of social network. However, those methods also have problems of the difficulty of understanding results and the high cost. The MAT based methods are proposed for solving the readability problem of NL methods. Those methods can easily grasp the node and the path of node on the network more than NL based approaches. But the MAT based methods waste a lot of memory (*i.e.* cost problem) because the visualization of matrix form represents sparse matrix on memory. Besides, the methods are also poor for path-finding tasks and understanding the results [1, 3, 4]. The hybrid based approaches also showed that makes relationship between users difficult to understand [1-5]. The typical approaches of hybrid based methods are MatLink [4] and MatTrix [5]. MatLink is proposed to resolve the problem of path-finding of the NL and MAT based approaches. This method enhances the performance of the NL and MAT based methods, since the edge of NL based method connects the node of MAT based method. However, this method also is difficult to understand of visualization results, since it have a disadvantage of complex connection between matrix and link [3-5]. MatTrix method is proposed for solving the disadvantage of MatLink method regarding the readability. This method uses the small size multi-matrix to connect link and node. This method is also difficult to understand the nodes relationship since it have a matrix based methods. In our previous works, we proposed the visualization methods using the fuzzy relational product and the correlation matrix to represent hierarchy tree form with respect to node and link of social network. Our methods resolve the readability problem [6, 7]. However the methods have a problem of the computational complexity for increasing at the social networks data.

Most of social network visualization methods have been only focusing on presentation of social network relationship. However, the methods do not consider an efficient processing speed and computational complexity for increasing at the ratio of arithmetical of a big data regarding social networks. This paper proposes a cloud based on visualization method to visualize an inherence relationship of user on social network.

In order to resolve the limitation of the social network visualization methods, this paper proposes a cloud based on visualization method to visualize an inherence relationship of user on social network which uses internal relation of the correlation matrix and the external relation of network traffic quantity. The proposed method can intuitionally understand the user's social relationship since the method uses correlation matrix to represent a hierarchical relationship of user nodes of social network. It also can easily identify a key role relationship of users on social network. In addition, the method uses Hadoop based on cloud for distributed parallel processing of visualization algorithm, which it can expedite the big data of social network.

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures [8].

This paper is organized as follows: Section 2 reviews the related works regarding correlation matrix; Section 3 presents the proposed method; Section 4 shows the performance

evaluation and experimental results of the proposed method. Finally, in Section 5 concludes this paper.

2. Correlation Matrix

In this section, we give a brief introduction to correlation matrix that is used to take internal relation of the user on social network. The key concepts of correlation matrix can be formalized as follows [9, 10].

Definition) Let $M=[m_{ij}]$ be a term-document matrix with t rows and N columns, where $m_{ij} = w_{i,j}$, *i.e.*, each entry ij in the matrix is given by the weight associated with the term-document pair (k_i, d_j). Given that M^T is the transpose of M, the matrix $C = M \cdot M^T$ is a term-term correlation matrix. Each element $c_{u,v} \in C$ expresses a correlation between terms k_u and k_v, given by

$$c_{u,v} = \sum_{d_j} w_{u,j} \times w_{v,j} \tag{1}$$

The term-term correlation matrix C establishes a relationship between any two terms k_u and k_v, based on their joint co-occurrences inside documents of the collection, as exemplified in Figure 1. This relationship is quantified by the correlation factor $c_{u,v}$. the higher the number of documents in which the terms k_u and k_v co-occur, the stronger is this correlation [9].

$$\begin{matrix} & d_1 & d_2 \\ k_1 & w_{1,1} & w_{1,2} \\ k_2 & w_{2,1} & w_{2,2} \\ k_3 & w_{3,1} & w_{3,2} \end{matrix}$$

(a) term-document matrix M

$$\begin{matrix} & k_1 & k_2 & k_3 \\ d_1 & w_{1,1} & w_{2,1} & w_{3,1} \\ d_2 & w_{1,2} & w_{2,2} & w_{3,2} \end{matrix}$$

(b) M's transpose matrix M^T

$$\begin{matrix} & k_1 & k_2 & k_3 \\ k_1 & w_{1,1}w_{1,1} + w_{1,2}w_{1,2} & w_{11}w_{2,1} + w_{1,2}w_{2,2} & w_{1,1}w_{3,1} + w_{1,2}w_{3,2} \\ k_2 & w_{2,1}w_{1,1} + w_{2,2}w_{1,2} & w_{2,1}w_{2,1} + w_{2,2}w_{2,2} & w_{2,1}w_{3,1} + w_{2,2}w_{3,2} \\ k_3 & w_{3,1}w_{1,1} + w_{3,2}w_{1,2} & w_{3,1}w_{2,1} + w_{3,2}w_{2,2} & w_{3,1}w_{3,1} + w_{3,2}w_{3,2} \end{matrix}$$

(c) $M \times M^T$ term-term correlation matrix

Figure 1. Correlation matrix for a sample collection with just two document and tree terms [9]

3. Proposed Method

This paper proposes a visualization method to visualize an inherence relationship of social network user based on Hadoop which uses internal relation of the correlation matrix and the external relation of network traffic quantity. The proposed method consists of three phases: preprocessing, calculating visualization relationship, and visualizing user relation, as shown in Figure 1. In the subsection below, each phase is explained in full.

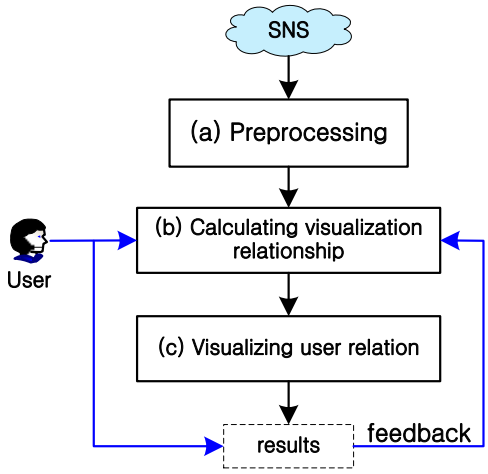

Figure 2. Visualization method based on cloud technique

3.1. Preprocessing

In preprocessing phase of Figure 2(a), SNS information is obtained for next phase of calculating visualization relationship. This paper explains an example to apply our proposed method using a structure of Twitter. Table 1 shows the obtained information from a structure of Twitter. The obtained information is internal relation of the correlation matrix and the external relation of network traffic quantity. The preprocessing phase consists of two steps.

First step preprocess the contents of Twitter for internal relation of the correlation matrix as follows. After the given obtainment results of Tweet are decomposed into individual users, the stop words are removed using Rijsbergen's stop words list, and word stemming is removed using Porter's stemming algorithm [10]. Then the term user frequency matrix T is constructed from the users set. Let T be m by n terms by users matrix, where m is the number of terms and n is the number of user. The element of matrix T, $T_{i,j}$ represents i'th term frequency of j'th user [10].

In second step, the information of network topology of Twitter is extracted for external relation of network traffic quantity.

Table 1. Information structure of Twitter

	item	description
Profile	Name *u*	Tweet user
Tweet	Tweet *tw*	Text-based posts of up to 140 characters
	day *d*	Posting day
Mention	Sender u_a	User of sending message
	Receiver u_b	User of receiving message
Retweet	Retweetee *ru*	Referenced user of tweet
	Retweeter *su*	User refer to tweet

3.2. Calculating visualization relationship

The visualization relationship is calculating using Hadoop [8] for distributed parallel processing. In the calculating visualization relationship phase in Figure 2(b), this phase consists of internal relation of the correlation matrix and the external relation of network traffic quantity for constructing hierarchy relationship of SNS. The internal relation of the correlation matrix represents the relationship between users to be derived from the contents of user's post. The external relation of network traffic quantity means the relationship between users on social network topology.

The internal relation of the correlation matrix is derived from user-user correlation matrix. User-user correlation matrix is calculated by equation (1) and user-term frequency matrix. Equation (1) is converted to MapReduce operations of Hadoop for distributed parallel processing (*e.g.*, $Z_1 = w_{u,j} \times w_{v,j}$ is computed to Map and Reduce).

MapReduce is a programming model and associated infrastructure that provide automatic and reliable parallelization once a computation task is expressed as a series of Map and Reduce operation. Open-source implementations of MapReduce infrastructure are readily available such as the Apache Hadoop project [8].

The The external relation of network traffic quantity considers directional characteristics of user relationship, mention period, amount of message, etc. The external relation, *e*( ), is given by as follows,

$$\mathrm{e}(u_a \rightarrow u_b) = \left\{\left(\frac{ntw \times nt}{ttw}\right) \times \frac{d}{\sum_{i=1}^{d}(d_i-(1-i))}\right\} + \left(\frac{nru}{ttw+nsu}\right) \quad (2)$$

where, u_a and u_b are the *a*'th and the *b*'th users, respectively. A right arrow, →, is the direction from sender to receiver, *ntw* is the number of mention message from *a* to *b*, *nt* is the number of term of all mention tweet between *a* and b. *ttw* is the number of mention tweet of all user on social network, *d* is during of posting day of referenced message, *nru* is the number of retweetee, *nsu* is the number of user refer to tweet.

Equation (2) is also converted to MapReduce operations of Hadoop for distributed parallel processing. Table 2 shows the summary of distributed parallel processing for external relation using MapReduce.

Table 2. Summary of external relation MapReduce operations

Stage	$\mathrm{e}(u_a \rightarrow u_b)$
1	$W_1 = \left(\frac{ntw \times nt}{ttw}\right)$ is computed to Map and Reduce
2	$X_1 = \frac{d}{\sum_{i=1}^{d}(d_i-(1-i))}$ is computed to Map and Reduce
3	$Y_1 = \frac{nru}{ttw+nsu}$ is computed to Map and Reduce
4	$Z_2 = (W_1 \times X_1) + Y_1$ is computed to Map and Reduce

Sum of internal relation and external relation, *sr*( ), is as follows,

$$sr(u_a \rightarrow u_b) = nor\left(c_{a,b}\right) + nor(e(u_a \rightarrow u_b)) \quad (3)$$

where, *nor*( ) is a normalization function. Table 3 shows the summary of distributed parallel processing for sum of internal and external relation using MapReduce.

Table 3. Summary of sum of internal and external relation MapReduce operations

Stage	$sr(u_a \rightarrow u_b)$
1	$W_2 = nor(c_{a,b})$ is computed to Map and Reduce
2	$X_2 = nor(e(u_a \rightarrow u_b))$ is computed to Map and Reduce
3	$Z_3 = W_2 + X_2$ is computed to Map and Reduce

3.4. Visualizing user relation

In this section, user relationship is visualized by constructing hierarchy relationship between users of social network to use the internal relation and the external relation for analysis of SNS. The visualization of hierarchy relationship of user's sociality improves the performance of analysis, because it reflects the internal relationship of user's posts by the correlation matrix and the external relation of user's social topology by user's activity.

Visualizing user relation as in Figure 2(c) can be described as follows. In the first step, the internal relation and the external relation are calculated by Equation (1) and Equaion (2) respectively. And the sum of the internal and the external relation is calculated by Equation (3). In the second step, the elements of sum of the internal and the external relation lower than cut value of Equation (4) are transformed to 0, and the others to 1. The hierarchy relationship is created by using the transformed results. In the final step, the hierarchy relationship of social network user transform to JSON form for visualizing on the web browsers by D3. JSON is a text-based open standard designed for human-readable data interchange [11]. D3 is a JavaScript library that uses digital data to drive the creation and control of dynamic and interactive graphical forms which run in web browsers [12]. In addition, user can adjust the hierarchy relationship of social network by changing the cut value, which enhances the analysis performance of SNS.
Cut value of the relation, *cv*( ), is as follows,

$$\mathrm{cv}(u_a) = \frac{\sum_{j=1}^{n} u_{a,j}}{n} \tag{4}$$

where, *n* is a number of total user on social network.

4. Experiment and results

The performance evaluation was conducted by comparing the proposed method (*i.e.* HR; hierarchy relation) with four methods (*i.e.*, NL [1], MAT [2], MatLink [4]) using the real social networks data [13-16]. NL denotes node-link based method, MAT denotes matrix graph based method, and MatLink denotes a hybrid method based on NL and MAT.

The mid-level readability tasks [3] used to measure the visualization performance. The tasks consists of evaluating connectivity (task 1, 2 and 3), finding central actors (task 4), and identifying communities (task 5). The readability of a node representation can be defined as the relative ease with which the user finds the information he is looking for [1]. The answer correctness is scored on a scale of 0 ~ 3, with 0 meaning error and 3 as the best answer [3]. In our experiment results, the average score of HRV is 18.32% higher than that of NL, 16.02% higher than that of MAT, 12.980% higher than that of MatLink. Figure 3 shows the result of visualization performance evaluation using the mid-level readability tasks.

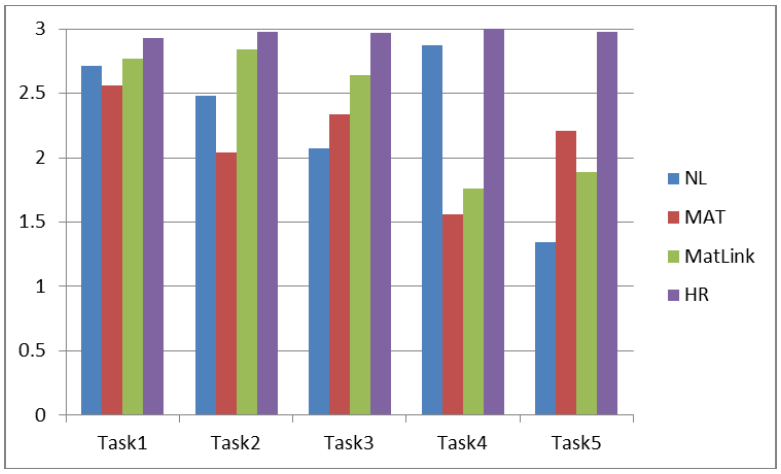

Figure 3. Evaluation result using mid-level readability tasks

5. Conclusion

In generally, social network visualization methods use a node graph based and a matrix based approaches. However, these approaches are difficult to understand a relationship between users on SN, because the user's interaction is large, complex and continuously changing. In addition, the methods do not consider an efficient processing speed and computational complexity for increasing at the ratio of arithmetical of a big data regarding social network. In order to overcome this limitation, this paper proposes a Hadoop cloud based on visualization method to visualize an inherence relationship of user on social network. The proposed method can intuitionally understand the user's social relationship since the method uses correlation matrix to represent a hierarchical relationship of user nodes of social network. It also can easily identify a key role relationship of users on social network. In addition, the method uses Hadoop based on cloud for distributed parallel processing of visualization algorithm, which it can expedite the big data of social network.

Acknowledgements

These should be brief and placed at the end of the text before the references.

References

[1] N. Henry and J. -D. Fekete, "MatLink: Enhanced Matrix Visualization for Analyzing Social Networks", LNCS 4663, Part II, **(2007)**, pp. 288-302.
[2] M. Ghoniem, J. D. Fekete and P. Castagliola, "On the readability of graphs using node-link and matrix based representations, a controlled experiment and statistical analysis", Information Visualization, vol. 4, no. 2, **(2005)**, pp. 114-143.
[3] S. Wasserman and K. Faust, "Social Network Analysis", Cambridge University Press, Combridge, **(1994)**.
[4] N. Henry and J. D. Fekete, "MatLink: Enhanced Matrix Visualization for Analyzing Social Networks", LNCS 4663, Part II, **(2007)**, pp. 288-302.
[5] N. Henry, J. D. Fekete and M. J. McGuffin, "NodeTrix: a Hybrid Visualization of Social Networks", IEEE Transactions on Visualization and Computer Graphics, vol. 13 Issue 6, **(2007)**, pp. 1302-1309.
[6] S. Park, J. J. G. Jeong, M. S. Yoe and S. R. Lee, "Visualization method of User Hierarchy of among SNS users", Journal of the Korea institute of Information and Communication Engineering, vol. 16, no. 8, **(2012)**, pp. 1717-1724.
[7] S. Park, J. W. Kwon, M. A. Jeong, Y. W. Lee and S. R. Lee, "Hierarchy Visualization method of SNS User Fuzzy Relational", The Institute of Electronics Engineering of Korea, vol. 49, no. 9, **(2012)**, pp. 76-84.
[8] Hadoop, http://hadoop.apache.org/, **(2013)**.

[9] B. Y. Ricardo, "Modern Information Retrieval: the concepts and technology behind search second edition", Addison Wesly, **(2011)**.
[10] W. B. Frankes and R. Baeza-Yaes, "Information Retrieval: Data Structure & Algorithms", Prentice-Hall, **(1992)**.
[11] JSON, http://en.wikipedia.org/wiki/JSON, **(2013)**.
[12] D3, http://en.wikipedia.org/wiki/D3.js, **(2013)**.
[13] Social Network Generation, http://www.infovis-wiki.net/index.php/Social_Network_Generation, **(2013)**.
[14] J. J. Jo and Y. C. Kim, "An Adaptive Pointing and Correction Algorith using Genetic Approach", International Journal of Multimedia and Ubiquitous Engineering, vol. 8, no. 6, **(2013)**, pp. 1-10.
[15] D. H. Suh and J. Yook, "An Evaluation Method of Event Information for Efficient Data Fusion", International Journal of Multimedia and Ubiquitous Engineering, vol. 8, no. 6, **(2013)**, pp. 59-66.
[16] K. Bae and H. G. Kim, "Optimal Point Correspondence for Image Registration in 2D Images", International Journal of Multimedia and Ubiquitous Engineering, vol. 8, no. 6, **(2013)**, pp. 127-140.

Authors

Yong-Il Kim

He received a B.S. in computer science from Chonnam University in 1984, and M.S. degree in computer science from Korea Advanced Institute Science and Technology (KAIST), Korea in 1986. From March 2002, he has joined as an Associate Professor at the Honam University, Gwangju, Korea. His research interests in data mining, big data, and intelligent agents. He is a member of IEEE.

Yoo-Kang Ji

He is a Visiting Professor at Dept. of Information & communication Eng., Dongshin Univ., South KOREA. He received the B.S., M.S., and Ph.D. degree in the Dept. of Information & Communication Eng. from DongShin Univ., KOREA in 2000, 2002 and 2006 respectively. He has worked professor in Dept. of Information & Communication Eng. DongShuin Univ. Mar. 2006 to Aug. 2009 His research interests in Mobile S/W, Networked Video and Embedded System.

Sun Park

He is a research professor at school of Information Communication Engineering at Gwangju Institute of Science and Technology (GIST), South Korea. He received the Ph.D degree in Computer & Information Engineering from Inha University, South Korea, in 2007, the M.S. degree in Information & Communication Engineering from Hannam University, Korea, in 2001, and the B.S. degree in Computer Engineering from Jeonju University, Korea, in 1996. Prior to becoming a researcher at GIST, he has worked as a research professor at Mokpo National University, a postdoctoral at Chonbuk National University, and professor in Dept. of Computer Engineering, Honam University, South Korea. His research interests include Data Mining, Information Retrieval, Information Summarization, Convergence IT and Marine, IoT, and Cloud.

