September 2014, Volume 2
Proceeding of 3rd International Conference on Computation for Science and Technology, ISSN xxxx-yyyy

Reducing Computational Complexity of Network Analysis using Graph Compression Method for Brand Awareness Effort

Andry Alamsyah[1,2*], Yahya Peranginangin[1], Budi Rahardjo[2], Intan Muchtadi-Alamsyah[3], Kuspriyanto[2]

[1]*School of Economics and Business, Telkom University, Jalan Telekomunikasi 1, Bandung, Indonesia.*
[2]*School of Electrical Engineering and Informatics, Bandung Institute of Technology, Jalan Ganesha 10, Bandung, Indonesia.*
[3]*Faculty of Mathematics and Natural Sciences, Bandung Institute of Technology, Jalan Ganesha 10, Bandung, Indonesia.*

Received: 20 September 2014 / Accepted: 30 November 2014

Abstract:
Online social media provides platform for social interactions. This platform produce large-scale data generated mostly from online conversations. Network analysis can help us to mine knowledge and pattern from the relationship between actors inside the network. This approach has been crucial in supporting prediction and decision-making process. In marketing context such as branding effort, using large-scale conversation data is cheaper, faster and reliable comparing mainstream approaches such as questionnaire and sampling. Social network analysis provides several metrics, which was built with no scalability in minds, thus it is computationally exhaustive. Some metrics such as centrality and community detections has exponential time and space complexity. With the availability of cheap but large-scale data, our challenge is how to measure social interactions based on those large-scale data. In this paper, we present our approach to reduce the computational complexity of social network analysis metrics based on graph compression method to solve real world brand awareness effort.

Key words: *Brand awareness, computational complexity, graph compression, large-scale data, social network, centrality*

Introduction

Today, our daily conversation in online social network services has contributed to the production of large-scale data. It is estimated that the volume of data production exceed 2,5 exabytes every day from online activities and the number is expected to double every 40 month [1]. Our social conversation in social media, user generated content, mobile application, real-time interaction, and trusted information, all contributed to production of large-scale data [2]. The large-scale data is a part of Big Data term which has emerged from large, fast, and complex data. Many of current state-of-the-art data technology contribute to solve Big Data problem [3], such as in Oil & Gas Mining Exploration [4], in Gnome / GenBank Project [5], in Astronomy [6], in Economic and Country Development [1,7].

The availibility of conversation data creates new unprecedented important opportunity, such as data patterns discovery to support decision-making process [8]. The insight we extract from online conversation are considering cheaper than having result from conventional data collection efforts such as questionaire and asking respondents. In conversational network, we often found various types of relations between two peoples and many types relations inside a simple network with limited number of peoples. This will be a shortcoming to the effort of capturing the overall picture and the network complexity, if we can only gathers limited amounts of data. We also have problem with limited scale of the conventional approach, typically hundreds peoples in one study, with the main issues on accuracy, time consuming and expensive process [9].

The relationship and connections among actors in online conversation can be modeled by using graph theory, where actors represented by vertex and relationship between actors represented by edges. In the real-world application, the model forms a complex network [10]. It is recognized that the topology and evolution of real-world network are governed by robust organizing principle such as random graphs, small world, preferential attachment and scale free networks. The combination between topology and robustness are crucial to the success and failures of network from challange and attack. The Social Network Analysis (SNA) provide several metrics to quantify social network such as centrality, community detection, homophilly, clustering coefficient, clique, mutuality, transitivity and some others. Having the ability to quantify the social network, many research and practition are dedicated themselves to develop graph mining technique [11].

Brand awareness is a branch in marketing effort to increase market awareness to the products or services brand. Brand awareness is the base step of brand equity that is defined as the value of having well-known brand. The most important step in brand awareness effort is defined on how a product is recognized by potential customer and associated with corrent product.

***Corresponding author:** Andry Alamsyah,
E-mail: andry.alamsyah@gmail.com

Brand awareness in online social media conversation mostly generated using advertising, thematic effort with relevant issue, and word-of-mouth in spreading information [12]. We know two important things regarding brand awareness using social network model [13]. First, the scenario brand awareness spreading process that similar to disseminate awareness through advertising and word-of-mouth. Second, we achieve brand awareness process more effective and efficient by understanding network formations that leads to social behavior of market.

We have described the advantages, opportunities, model and metrics. Although it is very promising, there are still some challenges to implement the process into practical world. First challenge is to identify that we have the right data or have the right people to unearth the insight from such large and complex network data. Second, connectivity and data access problem, since majority of data points are not yet connected to the network. Third, the fast evolve of technology landscape in data world means the needs of strong and innovative methodology that can adapt to changes. Fourth, in the computation side, we have limited processing powers and memory. The last challenge is we deal with different ownership of data where security issue can be a problem.

Network Model and Complexity Problem

We model the conversation in graph. Graph is defined as set of $G=(V,E)$ where V is set of vertices and E is set of edges. In random social network, the number of edges is bigger than the number vertices, which can be formalized as $E \subseteq [V]^2$ or $E \subseteq \begin{pmatrix} V \\ 2 \end{pmatrix} = \frac{V!}{2!(V-2)!}$ [14].

In Figure 1, we have graph representation of 7 actors and 5 relationships.   is the number of vertices in graph G, while   is the number of edges. The degree of a vertex v, written as d(v) means the number of edges   connected to v. Those are the basic of graph properties, there are many more complex attributes to represent real-world network such as: path, cycles, walk, connectivity, trees, matching and some other characteristics.

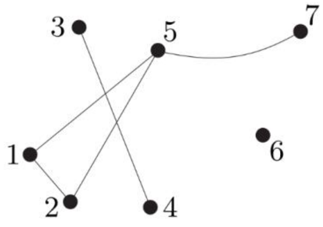

Figure 1. Graph G(V,E) with set of vertices V = {1,2,3,4,5,6,7} and set of edges E = {{1,2},{1,5},{2,5},{3,4},{5,7}}.

To understand the computational complexity in the next explanation, we take example from *centrality* metric. These metric measures are the most influential vertices or actors in the network. There are at least four types of centralities. They are based on connection, diameter, flow and line-ranks. The connection-based measures centrality of vertex based on the number of others vertices directly connected to them. This metric called degree centrality $C_D(i)$, the higher value of degree centrality in vertex i means the more connected vertex i to the rest of the network. The diameter-based centrality measures the average distance of vertex in question to any vertices in the network. The diameter-based connection centrality called closeness centrality $C_C(i)$. The higher value of closeness centrality means the shorter distance or the faster vertex i to reach others vertices in the network. The flow-based centrality called betweenness centrality $C_B(i)$ measure the number of shortest path going through vertex i when connecting any pair of vertices in the network. The higher value of betweenness centrality means the higher likelihood a vertex becomes bridge / hub connecting different group / component of the network. The formal methods of each centrality metric are shown in Table 1.

Table 1. Centrality metric

	Name	Description and Formal Method
1	*Degree Centrality*	*Measure the number of connection of a vertex* $C_D(i) = \sum_{o \le j \le n} k_j(i)$, $k_j(i)$ *is* j^{th} *degree of vertex i*
2	*Betweenness Centrality*	*Measure the number of shortest path between any two vertices through a vertex in question* $C_B(i) = \sum_{i \neq s \neq t} \frac{\sigma_{st}(i)}{\sigma_{st}}$, $\sigma_{st}(i)$ *is the number of shortest path between vertex s and t through vertex i*
3	*Closeness Centrality*	*Measure the average distance between a vertex in question to any vertices in the network* $C_C(i) = \sum_{i \neq j} \frac{n-1}{d_{ij}}$, d_{ij} *is the distance between vertex i and vertex j*
4	*Eigenvector Centrality*	*Measure proportional value of each vertex to total weight of neighborhood vertices.* $C_E(i) = \frac{1}{\lambda} \sum_j A_{ij} j$, λ *is the eigen value,* A_{ij} *is the value*

The illustration on how complexity increases when we deal with large network can be explains as follows. We look more details to the operation on closeness centrality

$C_C(i)$ with the help of illustration in Figure 2. $C_C(4)$ is closeness centrality on vertex 4, to get this value we need to compute at least number of hops to the other *n-1* vertices. The complexity increase the further the distance as we need to check whether the currect choice distance is the shortest one, if not we need to find the shortest alternative. We also need to compute $C_C(i)$ on each vertex and rank them, in this graph we found $C_C(4)$ is the highest closeness centrality with value 0.62. In betweenness centrality $C_B(i)$, we first compute the shortest path between any pair of vertices and check whether the path pass through vertex *i*. We iterate the process for all pair of vertices in network and we rank them. In this graph we found $C_B(4)$ is the highest betweenness centrality with value 0.54.

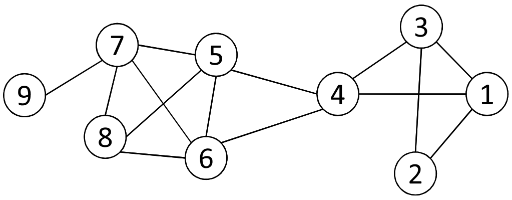

Figure 2. Illustration of $C_C(i)$ and $C_B(i)$ computation.

It is clear from the description above that the algorithm to compute $C_C(i)$ and $C_B(i)$ are computationally exhaustive. The fastest algorithm need time complexity $O(n^3)$ and space complexity $O(n^2)$, with n is the number of vertices [15]. There are efforts to reduce complexity using several strategies such as social network characteristic of lower density graph [15], estimation value using random selection pivot [16], strategy for rank top-k value [17], and parallel computation [18][19]. The main issues of proposed efforts are the limited scale of network / graph tested (around 10000 vertices). In general, there are several options to reduce graph size such as *graph compression, random or strategic vertices sampling, model transformation* and *features selection*. In this paper, we proposed new strategy to reduce large-scale graph using graph compression algorithm [20].

Graph Compression

Graph compression works based on *Minimum Description Length* (MDL) [21] rules, which outline the need to represent the message using less number of data, without altering the meaning of the message. MDL reflect that the best representation is the one with the minimum cost to form it. We have two choices implementing graph compression:

1. MDL representation which exact representation of the original graph
2. Approximation representation which representation that allowing controllable errors for gaining faster processing time

The MDL representation of graph $G = (V_G, E_G)$ is $R = (S,C)$, which contain graph summary $S = (V_S, E_S)$ and a set of edge-correction *C*. Each vertex *v* in V_G is a member of supervertex *V* in V_S which represent set of vertices in *G*. A superedge $E = (V_i, V_j)$ in E_S represent set of all edges who connecting all pairs vertices in V_i and V_j. Edge correction *C* has two member *+e* dan *–e* , which means we can adding or delete edges when we reconstruct the original graph. The visualisation process can be seen at Figure 3 [20]. On the left hand side is the original graph *G*, and on the right hand side is the MDL representation *R*. The formations of supervertex *V* in *R* from vertices are based on the minimum cost reduction. The important boundary to do the computation iteration is the cost reduction, which symbolize by $s(u,v) = (c_u + c_v - c_{uv}) / (c_u + c_v)$ where c_u cost representation of vertex *u*.

The approximation representation of graph $G = (V_G, E_G)$ with error ε , $0 \leq \varepsilon \leq 1$ is $R_\varepsilon = (S_\varepsilon , C_\varepsilon)$. This representation should qualify that each vertex $v \in G$, *error (v)* = $|N_v\text{'}- N_v|+|N_v - N_v\text{'}| \leq \varepsilon|N_v|$, where N_v *and* N_v*'* are set of neighborhood vertices of *v* in graph G and R_ε. $|N_v\text{'}- N_v|$ is the number of edges that exist in the graph representation but not on the original one, the opposite symbol is the vice versa. The simple way to construct approximation representation is by removing some edges from *C* dan *S* on MDL representation, as long as the actions do not violate bounded error ε. In other words, we only need to check whether edge *e = (u,v)* comply to the rule that at least $(1 - \varepsilon)$ of the original edge still connecting *u* and *v*.

In this paper we focus our work on MDL representation. There are two types of MDL representation, the exact representation and the randomized representation. The difference of the two approach is on the step of updating the highest overall cost reduction in exact representation, while in randomized representation the updating process are done by choose the closest / local vertices who gives the highest cost reduction. In this paper, we do not test the approximate representation yet, although it is very promising but we need more time until we have the conclusive result. The algorithma of exact MDL and randomized MDL representation based on [20] can be seen below.

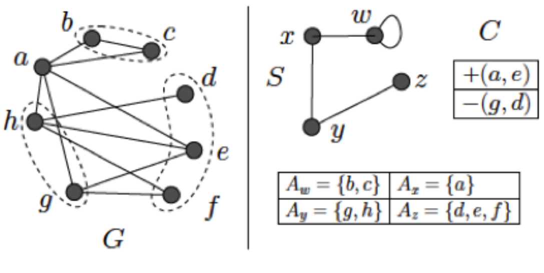

Figure 3. The original graph G (left) and the MDL representation R (right).

Exact MDL Representation Algorithm

```
/*Initialization phase*/
$V_S$=$V_G$; H= Ø;
for all pairs (u,v) ∈$V_S$ that are 2 hops apart do
    if (s(u,v) > 0) then insert (u,v,s(u,v)) into H;
/*Iterative merging phase*/
while H ≠ Ø do
    Choose pair (u,v) ∈H with largest s(u,v) value;
    w = u ∪ v; /*merge u and v into supernode w*/
    $V_S = V_S – \{u,v\} ∪ \{w\}$;
    for all x ∈$V_S$ that are within 2 hops of u and v do
        Delete (u,x) or (v,x) from H;
        if s(w,x) > 0 then
            insert (w,x,s(w,x)) into H;
    for all pairs (x,y), such that x or y is in $N_W$ do
        Delete (x,y) from H;
        if s(x,y) > 0 then
            insert (x,y,s(x,y)) into H;
/*Output phase*/
$E_S$ = C = Ø;
for all paris (u,v) such that u,v ∈ $V_S$ do
    if ($A_{uv}$ > ( |$\Pi_{uv}$|+1)/2) then
        Add (u,v) to $E_S$;
        Add –(a,b) to C for all (a,b) ∈ $\Pi_{uv} – A_{uv}$;
    else
        Add +(a,b) to C for all (a, b) ∈ $A_{uv}$;
return representation R=(S=($V_S$.$E_S$),C)
```

Randomized MDL Representation Algorithm

```
U = $V_S$ = $V_G$; F= Ø;
while U ≠ Ø do
    Pick a node u randomly from U;
    Find the node v with the largest value of s(u,v) within 2 hops of u;
    if (su,v) > 0) then
        w = u ∪ v;
        U = U - {u,v} ∪ {w};
        $V_S = V_S – \{u,v\} ∪ \{w\}$;
    else
        Remove u form U and put it in F;
/*output phase same as Exact MDL*/
```

Experiments

We crawl various online conversation about different brands in Twitter, as the result we found different network size according to the conversation size. Twitter provide uniform format to mine the data and this is much more efficient than doing web mining. We found four different brand with network size as in the Table 2 as follows:

Table 2. The brand network information

	Name	Vertex	Edges
1	*Brand A*	335	2520
2	*Brand B*	7115	103689
3	*Brand C*	36392	183831
4	*Brand D*	75879	508837

We mentioned earlier the computational complexity of social network metric $C_C(i)$ and $C_B(i)$ are $O(n^3)$ for time complexity and $O(n^2)$ for space complexity. It is computationally exhaustive, for example the fastest metric computation is 3.7×10^7 times for the smallest network (Brand A). We apply the exact MDL representation and randomized MDL representation to brand network above. The experimentation results are shown in Table 3 and Table 4 below. The information including brand network name, number of original vertices and edges, number of compressed vertices and edges, original space and compressed space, and time needed to run the compression algorithm.

Table 3. Exact MDL algorithm result, (o) original, (c) compressed

Name	Vertex / Edges (o)	Vertex / Edges (c)	Space (o) / Space (c)	Time
Brand A	335 / 2520	181 / 356	2519 / 1481	2.048
Brand B	7115 / 103689	2984 / 11897	100762 / 81878	2242.110
Brand C	36392 / 183831	12737 / 25860	183831 / 127461	4376.689
Brand D	75879 / 508837	29910 / 34742	405740 / 328923	31076.442

Table 4. Randomized MDL algorithm result, (o) original, (c) compressed

Name	Vertex / Edges (o)	Vertex / Edges (c)	Space (o) / Space (c)	Time
Brand A	335 / 2520	165 / 253	2519 / 1558	2,048
Brand B	7115 / 103689	3094 / 9938	100762 / 84172	37,133
Brand C	36392 / 183831	12448 / 21037	183831 / 131960	89,420
Brand D	75879 / 508837	30400 / 29843	405740 / 334758	426,069

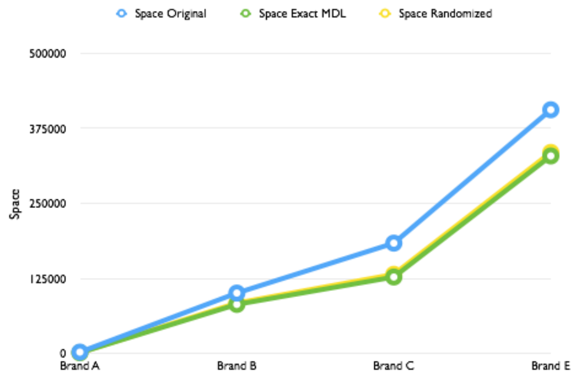

Figure 4. The space complexity of the original network, exact MDL representation and randomized MDL representation.

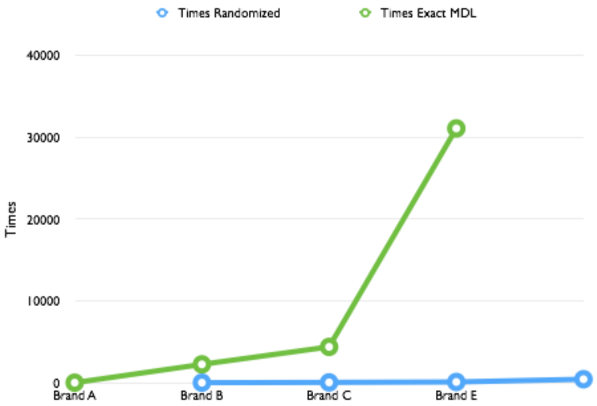

Figure 5. The time complexity of both exact MDL representation and randomized MDL representation.

It is shown in Figure 4 and Figure 5 that both exact MDL representation and randomized representation compress the size of original network in about 50%. By reducing the space complexity, it will significantly resulted in faster metric computation. Randomized MDL perform significantly faster than exact MDL representation in compressing the original network. Keep in mind that we have not yet compare time needed for the overall process of metric computation on social network graph and metric computation on compressed social network graph plus time needed for graph compression.

Conclusions

We have shown that graph compression can reduce time and space complexity of large-scale graph. The method is very promising to help the social network quantification, especially in marketing effort such as brand awareness, where large-scale conversation data is rarely to be used. This paper is the base of the more conclusive research. Our next step is check the accuracy of the method by comparing the social network metric value and ranks from the original and the compressed graph.

In the future, we can expand the research using other methodologies to increase computation speed, such as model transformation, random sampling/strategic samping and feature selection. We need to compare the efficiency of those methods with graph compression in social network case.

References

[1] A. MacAfee, E. Brynjolfsson. *Big Data: The Management Revolution.* In Harvard Business Review. October 2012

[2] D. Lazer, A. Pentland, L. Adamic, S. Aral, A.L. Barabasi, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, T. Jebara, G. King, M. Macy, D. Roy, and M. Van Alstyne, Computational Social Science, *In Science*, **323**(5915), 2009, 721-723.

[3] G. Cottlob, G. Grasso, D. Olteanu, and C. Schallhart, *Big Data: 29th British national conferences on database*, BNCOD 2013, Oxford, UK. July 2013.

[4] A. Kristensen, *How the oil and gas industry can gain value from Big Data?*, IBM Corporation, 2013.

[5] D. Benson, I. Karsch-Mizrachi, D.J. Lipman, J. Ostell, and D.L. Wheeler, Genbank. *Nucleic Acids Research*, **36** *(Database)*, 2008, D25–D30.

[6] K.D. Borne, *Astroinformatics: a 21st century approach to astronomy*, arXiv:0909.3982. 2009

[7] A. Zaslavsky, C. Perera, and D. Georgakopoulus, *Sensing as a Service and Big Data*, In Proceeding of the International Conference on Advances in Cloud Computing. Bangalore, India, July 2012

[8] D. Watts, *Annu. Rev. Sociol.*, **30**, 2004, 243.

[9] A. Alamsyah and Y. Peranginangin. Effective knowledge management using big data and social network analysis, *In Learning Organization: Management and Business International Journal*, **1**(1), 2013.

[10] R. Albert and A.L. Barabasi, Statistical mechanic of complex network, *In Reviews of Modern Physics*, **74**, 2002.

[11] S.U. Rehman, A.U. Khan, and S. Fong, *Graph Mining: A survey of graph mining techniques*, In 7th International Conference on Digital Information Management (ICDIM). Macau, 2012.

[12] D.A. Aaker, Measuring Brand Equity Accross Product and Markets, *California Management Review*, **38**(3), 1996, 102-120.

[13] A. Alamsyah, F. Putri and O.O. Sharif, *Proceedings of the 2014 ICOICT International Conference on Information and Communication Technology*, **83**, 2014.

[14] R. Diestel, *Graph Theory: Electronic Edition 2005*. Springer-Verlag Heidelberg, New York 1997, 2000, 2005

[15] U. Brandes, A faster algorithm for betweenness centrality, *In Journal of Mathematical Sociology*, **25**(2), 2001, 163-177.

[16] K. Okamoto, W. Chen, and X.Y. Li, Ranking of closeness centrality for large scale social network, *In Frontiers in Algorithmics, Lecture Notes in Computer Science*, **5059**, 2008, 186-195.

[17] U. Brandes and C. Pich, Centrality Estimation in large networks, *In International Journal Bifurcation and Chaos, Special Issue on Complex Network Structure and Dynamics*, **17**, 2007, 2303.

[18] D. Bader and K. Madduri, Parallel Algorithms for Evaluating Centrality Indices in Real-World Networks, *In*

Proceedings of IEEE International Conference on Parallel Processing, ICPP'06, 2006.

[19] G. Tan, D. Tu, and N. Sun. A Parallel Algorithm for Computing Betweenness Centrality, *In Proceedings of IEEE International Conference on Parallel Processing, ICPP'09*, 2009.

[20] S. Navlakha, R. Rastogi, and N. Shrivastava, Graph Summarization with Bounded Error. *In Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2008.

[21] P.D. Grunwald, *The Minimum Description Length Principle*, The MIT Press, 2007.