

A GRAPH-BASED ALGORITHM FOR INTERPERSONAL TIES CLUSTERING IN SIGNED NETWORKS

Sumalee SANGAMUANG

Abstract: Social ties are formed as a result of interactions and individual preferences of the people in a social network. There are two opposite types which are interpreted as friendship vs. enmity or trust vs. distrust between people. The aforementioned social network structure can be represented by a signed graph, where people are the graph's vertices and their interactions are graph's edges. The edges can be positive and negative signs. To determine trustworthiness, this paper considers the problem of a signed graph partitioning with minimizing the sum of the negative edge's weight and balanced size of its clusters. An efficient algorithm to solve such a problem is proposed. The experimental results show that the proposed algorithm outperforms in terms of the execution times and the accuracy within the given bounds.

Keywords: graph partition; min-cuts; signed graphs; social networks; social ties

1 INTRODUCTION

Recently, the social internet of things (SIoT) has been proposed and is the subject of a rapidly increasing research effort [1], [2] and [3]. The structure of SIoT has been formulated in [4] based on the notion of social relationships among the IoT objects. The aforementioned relationships can be represented by a signed graph where the vertices are the IoT objects and their interactions are the edges. The problem of service selection was discussed in [5] based on social influence. Given the set of IoT nodes providing the needed service, every node, after interaction with another node, gives a rating to the node based on the service it provides. All nodes interact among themselves based on a social network structure. Each object has to be provided with the best and most trusted reliable node.

In a social network, social ties are formed as a result of interactions and individual preferences of the people in a social network. One person may change to another in the causes of behavior related to reliability. There is a level of trustworthiness in a social network that can be determined by the PageRank [6-8]. Thus, the level of trustworthiness is considered from the values of PageRank incoming links. For real-world scenarios, there are two opposite types, i.e. there is trust versus distrust between the relationships. Thus, the edges should be the positive and negative signs.

Fig. 1 shows an example of a signed graph that represents the social relationships among IoT objects in a real-world scenario. In this figure, trust is represented by a straight line, while distrust is represented by a dotted line. The level of trustworthiness is represented by the edge's weight. If the edge's weight is small, there is not enough trustworthiness.

On the other hand, if the edge's weight is large, there are high levels of trustworthiness. The number of IoT devices and services is steadily increasing, and given the connectivity and reliability of a large signed chart of centralization systems, it is not efficient. A distributed system is a viable choice. In [5], a large-scale service selection problem was

considered. Based on such a problem, the large signed graph has to be partitioned into smaller clusters for reducing execution times.

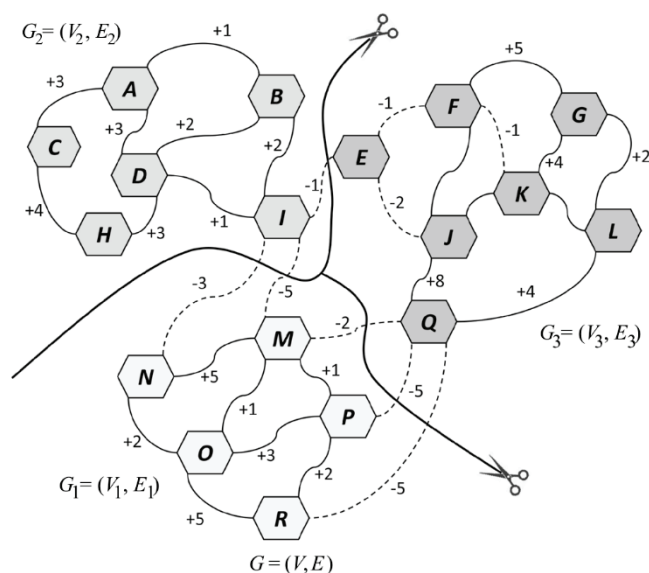


Figure 1 An example of the minimal partition of the signed graph

The set of clusters is called the partition of the signed graph. For each input graph, there is a number of possible partitions. This paper considers an optimization problem to find approximately the same cluster size that the sum of the weight of the negative external-edge's is minimized while the sum of the positive external-edges is less than the given constant. A graph-based algorithm for interpersonal ties clustering in signed networks is proposed to solve the aforementioned problem. The example of the optimal partition is shown in Fig. 1. Let integers $k = 3$ and $\omega = 5$ and $G = (V, E)$ be a signed graph. The signed graph is partitioned into its minimal partition $P_c = \{G_1, G_2, G_3\}$ where the sum of negative external edge's weight is -21 , as a minimized edge's weight, and the sum of positive external edge's weight is 0 (less than a given constant ω).

2 PROBLEM DEFINITION INTRODUCTION

In this section, the notations and problem statement are defined as follows.

Definition 1 (Signed Graph): Let $G = (V, E)$ be a signed graph which consists of n vertices. The vertices' set $V = \{v_1, v_2, v_3, v_4, \dots, v_n\}$, and the edges' set $E = \{(v_i, v_j, w_{ij}) \mid i \neq j, w_{ij} \in I\}$.

Definition 2 (Cluster): Given $G = (V, E)$, a graph $G_c = (V_c, E_c)$ is a cluster of G where $V_c \subseteq V$ and $E_c \subseteq E$.

Definition 3 (Partition): Given $G = (V, E)$, a set of its clusters $P_c = \{G_1, G_2, G_3, G_4, \dots, G_c\}$ is called a partition of G that c is the number of clusters in the partition.

Definition 4 (Internal Edge): Given $G = (V, E)$, and let P_c be its partition. For each cluster G_i in P_c , a pair $(v_i, v_j, w_{ij}) \in E_c$ is called an internal-edge. There are two types of an internal-edge, i.e., a negative internal-edge and a positive internal-edge. A set of a positive internal-edge is $E^+ = \{(v^+, v^+, w^+) \mid w^+ \in I^+\}$ while a set of a negative internal-edge is $E^- = \{(v^-, v^-, w^-) \mid w^- \in I^-\}$ that $E^+ \cup E^-$.

Definition 5 (External Edge): Given $G = (V, E)$, and let P_c be its partition. For each cluster G_i in P_c , a pair $(v_i, v_j, w_{ij}) \in E \cup_{c=1}^k E_c$ is called an external-edge.

Definition 6 (Cut): Given $G = (V, E)$, and let P_c be its partition. For each cluster G_i in P_c , a set of the external-edges is called a cut C which consists of a set of positive external-edges $C^+ = \{(v_i^+, v_j^+, w_{ij}) \mid w_{ij} \in I^+\}$ and a set of negative external edges $C^- = \{(v_i^-, v_j^-, w_{ij}) \mid w_{ij} \in I^-\}$.

This paper considers an optimization problem to find k a roughly equal size of the clusters that the sum of negative external-edge's weight is minimized, while the sum of positive external-edges is less than the given constant. For the problem instance, it is given $G = (V, E)$ and positive integers k and ω . The problem returns P_c which consists of k roughly equal size of the clusters, meanwhile, the sum of the edge's weight w_{ij} in C^- is minimized, and the sum of the edge's weight w_{ij} in C^+ is less than ω . This problem is the under category NP-hard by reducing to the min-cut max-flow problem.

3 THE ALGORITHM

In [9] and [10], an algorithm for partitioning a graph into balanced clusters with a minimal size of cuts was proposed. Based on the aforementioned algorithm, this section presents a heuristic algorithm to find balanced clusters whose sum of the negative external-edge's weight is minimized, while the sum of positive external-edges is less than a given constant.

Algorithm 1 separates a signed graph $G = (V, E)$ into its partition P_c with a satisfied condition at line 1. The sum of all edge's weights in C^+ is less than ω . The input graph is recurrently separating itself into a smaller cluster G_c until the size of P_c is k (line 9 to line 13).

In Line 9, the set of vertices of the input graph are separated into two subsets, $X \in V$ and $Y \in V$, with an equal size by using the Algorithm 2. Line 10 and line 11 create clusters from x_i and y_i . Thus, both clusters are equal to the

number of vertices and are contained within a partition of the input graph. This partition is called an exact bisection of the input graph. Since both clusters half-separate themselves at line 12 and line 13, the Algorithm 1 runs in $O(|V| \log |V|)$ time.

Algorithm 1 Find the Cluster

Require: A signed graph $G = (V, E)$ and constants k, ω .

Ensure: A partition P_c .

```

1: if  $f(C^+) < \omega$  &  $|P_c| < k$  then
2:   if  $P_c = \varnothing$  then
3:      $P_c \neq \varnothing$ 
4:   else
5:      $P_c \neq G_c$ 
6:   end if
7:   return  $P_c$ 
8: else
9:   find the cut-set  $\mathbb{C}_{V_i, V_j}$  using Algorithm 2
10:  create subgraphs  $G_i$  from  $V_i$ 
11:  create subgraphs  $G_j$  from  $V_j$ 
12:   $G_c \leftarrow FindCluster(G_i, \omega)$ 
13:   $G_c \leftarrow FindCluster(G_j, \omega)$ 
14: end if

```

Algorithm 2 is formulated based on the well-known heuristic algorithm which is called Kernighan-Lin [11]. The set of vertices V is separated into two disjoint subsets X and Y of equal or nearly equal size which minimizes the cost of crossing edges (the sum of the edge's weight). However, the Kernighan-Lin algorithm is limited with the negative edges of the signed graph.

For determining the cost, let C_I be the internal cost of $x \in X$, which is the sum of the edge's weight w_x between x and other nodes in X , i.e., $\sum_{w_x \in X} (w_x)^2$. Let C_E be the external cost of $x \in X$ which is the sum of the edge's weight $w_{x,y}$ between the nodes $x \in X$ and $y \in Y$, i.e., $\sum_{w_{x,y} \in X} (w_{x,y})^2$.

Therefore, the difference of the internal cost and the external cost is formulated as follows:

$$D_x = C_E - C_I \quad (1)$$

Moreover, the cost for an interchange between x and y is

$$g_{x,y} = D_x + C_y - 2C_{x,y} \quad (2)$$

The term $C_{x,y}$ is the cost of the possible edge between x and y .

The algorithm tries to find an optimal series of interchange operations between the elements, minimizing the sum of the external edge's weight between x and y , which maximizes $g_{x,y}$ and then an optimal bisection of the input graph will be produced when the algorithm is finished. Algorithm 2 runs in $O(|V| \log |V|)$ time.

Algorithm 2 Find the max cut

Require: Given $G = (V, E)$.
Ensure: The max-cut of G .
1: Randomly half-separates the set V into sets X and Y
2: **while** $g_{\max} \leq 0$ **do**
3: Compute D values for all $x_i \in X$ and $y_i \in Y$
4: Let g , X^*_i and Y^*_i be empty lists
5: Let $k \leftarrow |V|/2$
6: **for** $i \leftarrow 1$ to k **do**
7: Find cost for the interchange between x_i and y_i by Eq. (2)
8: Remove x_i and y_i from further consideration in this step
9: Add g to $g[i]$, x_i to $X^*[i]$, and y_i to $Y^*[i]$
10: Update D values for the elements of $X^*[i] = X \setminus x_i$ and $Y^*[i] = Y \setminus y_i$
11: Find index i of (x_i, y_i) which $g_{\max} = \max_{i=1}^k g[i]$
12: **end for**
13: **end while**
14: **if** $g_{\max} > 0$ **then**
15: Swap $X^*[i]$ with $Y^*[i]$
16: Set $X = X^*$ and $Y = Y^*$
17: **end if**

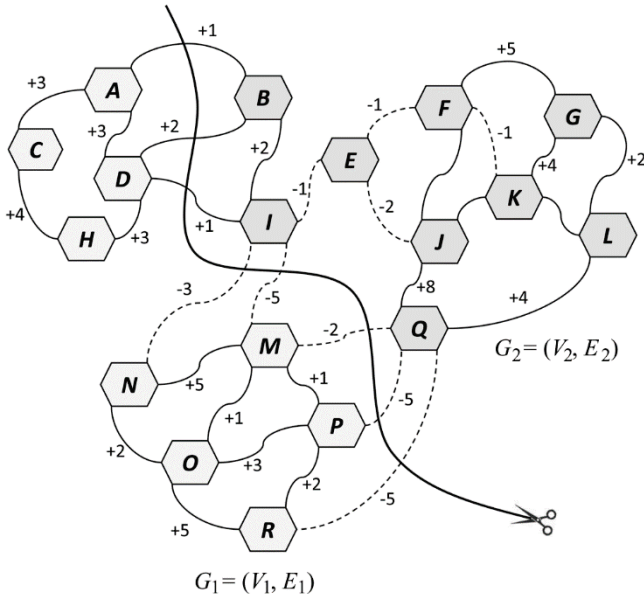


Figure 2 An example of the clusters of the input graph by using the Algorithm 2

Fig. 2 shows an example of the clusters of the input graph by using the Algorithm 2. Let $E = 5$, the input graph is separated into its partition which consists of the cluster G_i and the cluster G_j of equal sizes. Based on the heuristic approach, there are three positive external edges whose sum of the edge's weight is four. This partition can be acceptable because the sum of all edge's weight in C^+ is less than five.

4 EXPERIMENT EVALUATION

This section evaluates the performances, i.e., the partition's execution times and the accuracy. The experimental results show the performances of the proposed algorithm and MINOS solver by measuring the execution times and the accuracy to the partition. Moreover, the performances of the proposed algorithm on a real-world large-scale signed social network are evaluated.

The experiments use the datasets from Slashdot [12], a signed social network that contains friend/foe links between the Slashdot's users. This dataset was obtained in November 2008, November 2009 and February 2009 and it contains 77,350 users with 516,575 relationships, 81,871 users with 545,671 relationships and 82,144 users with 549,202 relationships.

Table 1 Performances in terms of execution times

V	Execution Times (min.)	
	Minos Solver	Proposed Algorithm
40	1.07	0.21
80	5.24	0.39
120	13.11	0.41
160	89.13	0.56
200	245.43	0.69

Table 2 Performances in terms of accuracy

V	Cuts	
	Minos Solver	Proposed Algorithm
40	16	100
80	21	152
120	23	203
160	38	240
200	52	298

Tab. 1 shows the execution times of the proposed algorithm and MINOS solver. If the graph sizes are increased, the execution times of MINOS solver will rapidly increase. On the other hand, the execution times of the proposed algorithm are less than one minute. For example, at the graph size of 200 vertices, the execution time of the MINOS solver is 245.43 minutes, while the execution time of the proposed algorithm is only 0.69 minute.

Tab. 2 shows the sizes of cuts from the partitions using the proposed algorithm and MINOS solver. The sizes of cuts from the partitions using the proposed algorithm are always higher. However, these solutions are ab acceptable region in the application for a social network analysis.

Tab. 1 and Tab. 2 present trade-off between the execution times and the accuracy. Since the MINOS optimizer is limited with solving very large-scale problems, the proposed algorithm is a viable choice. However, the execution times of the proposed algorithm had to be traded by its accuracy.

Tab. 3 shows the performances of the proposed algorithm on a real-word large-scale signed social networks, i.e. Slashdot in November 2008, Slashdot in November 2009 and Slashdot in February 2009. The experimental results show that the cuts of partitions from Slashdot in November 2008 are 1.14 percent, Slashdot in November 2009 are 1.93

percent and Slashdot in February 2009 are 2.17 percent, meanwhile, these execution times are less than one hour.

Table 3 Performances in terms of the proposed algorithm on large-scale networks

Signed Networks	Execution Times (min.)	Cuts
Slashdot (November 2008)	19.17	5873
Slashdot (November 2009)	37.62	10520
Slashdot (February 2009)	51.18	11905

For measuring the ranking accuracy, the Spearman's rank correlation coefficient [13] has been used for determining the degrees of the ranking's error. It is defined as $F(\sigma_1, \sigma_2) = \sum |\sigma_1(i) - \sigma_2(i)|$, where $\sigma_1(i)$ and $\sigma_2(i)$ are the rank of the vertex i in the centralized vertices ranking and distributed vertices ranking, respectively. This value indicates the error from the distributed vertices ranking, higher Spearman's footrule distance indicates higher degrees of the ranking's error.

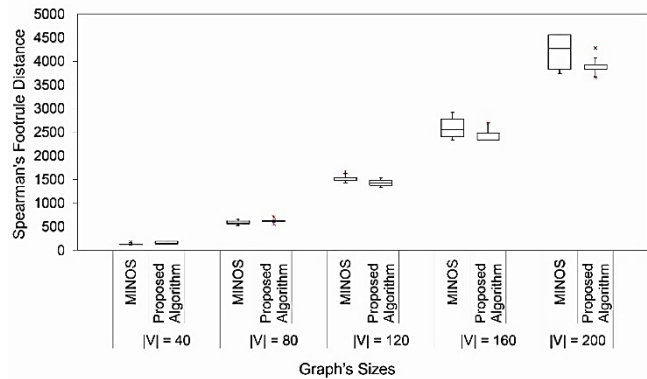


Figure 3 The ranking accuracy of the proposed algorithm compared with the MINOS solver

Fig. 3 shows the ranking accuracy of the proposed algorithm compared with the MINOS solver. The X-axis is the graph size in the ranges from 40 to 200, while the Y-axis is the Spearman's footrule distance. The proposed algorithm can perform faster when compared with the MINOS solver, while the accuracy is still acceptable for the vertices ranking on social networks.

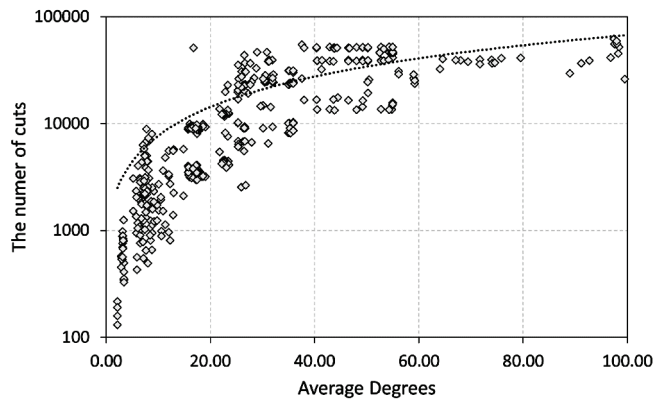


Figure 4 The cuts of the proposed algorithm

Fig. 4 shows impacts of the average degrees on the size in the ranges from 0 to 100, while the Y-axis is the size of the

cuts. This experiment uses 150 different randomly selected subgraphs of Slashdot in November 2008 to create 150 balanced partitions.

In this figure, if the average degrees are increased, the sizes of cuts will rapidly increase until the average degree is 30. However, the graph sizes gradually increase when the average degree is over 30. Likewise, Fig. 5 shows the impacts of the average degrees on execution times. It can clearly be seen that the execution times of the proposed algorithm are the logarithm of the average degrees of the graph.

Based on the experimental results, the proposed algorithm is very efficient for partitioning a large-scale signed social network.

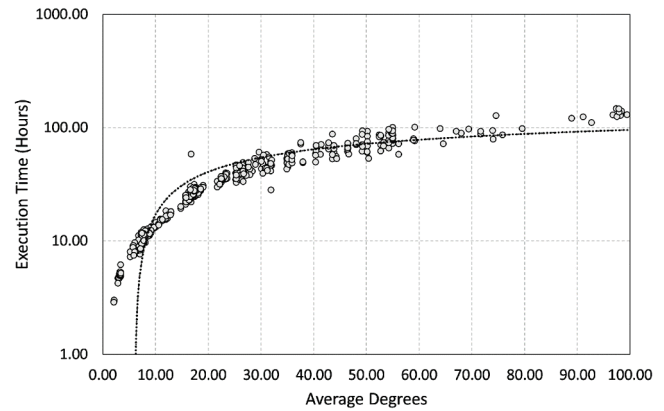


Figure 5 The execution times of the proposed algorithm

5 CONCLUSIONS

This paper considers an optimization problem to find roughly equal size of the clusters that the sum of negative external-edge's weight is minimized, while the sum of the positive external-edges is less than a given constant. A graph-based algorithm for interpersonal ties clustering in signed networks is proposed to solve the aforementioned problem. The experimental results show that the proposed algorithm can perform faster when compared to the MINOS solver, while the accuracy is still acceptable for the vertices ranking on social networks. Moreover, experimental results show that the number of cuts has an upper bound which will be considered as an approximation bound in future works.

Remark

The article was orally presented at the 23rd International Computer Science and Engineering Conference (ICSEC2019).

6 REFERENCES

- [1] Iera, A., Morabito, G., & Atzori, L. (2015, March). The social internet of things. In *2015 IEEE International Conference on Cloud Engineering*, 1-1. <https://doi.org/10.1109/IC2E.2015.68>
- [2] Nitti, M., Atzori, L., & Cvijikj, I. P. (2015, June). Friendship selection in the social internet of things: Challenges and

- possible strategies. *IEEE Internet of Things Journal*, 2(3), 240-247. <https://doi.org/10.1109/JIOT.2014.2384734>
- [3] Ortiz, A. M., Hussein, D., Park, S., Han, S. N., & Crespi, N. (2014, June). The cluster between internet of things and social networks: Review and research challenges. *IEEE Internet of Things Journal*, 1(3), 206-215. <https://doi.org/10.1109/JIOT.2014.2318835>
- [4] Atzori, L., Iera, A., & Morabito, G. (2011, November). Siot: Giving a social structure to the internet of things. *IEEE Communications Letters*, 15(11), 1193-1195. <https://doi.org/10.1109/LCOMM.2011.090911.111340>
- [5] Nizamkari, N. S. (2017, Jan). A graph-based trust-enhanced recommender system for service selection in IoT. In *2017 International Conference on Inventive Systems and Control (ICISC)*, 1-5. <https://doi.org/10.1109/ICISC.2017.8068714>
- [6] Chen, S. & He, K. (2015, Dec). Influence maximization on signed social networks with integrated PageRank. In *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, 289-292. <https://doi.org/10.1109/SmartCity.2015.86>
- [7] Hajian, B. & White, T. (2011, Oct). Modelling influence in a social network: Metrics and evaluation. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, 497-500. <https://doi.org/10.1109/PASSAT/SocialCom.2011.118>
- [8] Wu, J. (2017, March). Realtime personalized pagerank query for social network search. In *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 647-651. <https://doi.org/10.1109/IAEAC.2017.8054096>
- [9] Feige, U. & Krauthgamer, R. (2002, Apr). A polylogarithmic approximation of the minimum bisection. *SIAM J. Comput.*, 31(4), 1090-1118. <https://doi.org/10.1137/S0097539701387660>
- [10] Andreev, K. & Racke, H. (2006, November). Balanced graph partitioning. *Theory of Computing Systems*, 39, 929-939. <https://doi.org/10.1007/s00224-006-1350-7>
- [11] Kernighan, B. W. & Lin, S. (1970, Feb). An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 49(2), 291-307. <https://doi.org/10.1002/j.1538-7305.1970.tb01770.x>
- [12] Leskovec, J., Huttenlocher, D., & Kleinberg, J. (2010). Signed networks in social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ser. CHI '10*. New York, NY, USA: ACM, 1361-1370. <https://doi.org/10.1145/1753326.1753532>
- [13] Fagin, R., Kumar, R., & Sivakumar, D. (2003). Comparing top k lists. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 28-36. <https://doi.org/10.1137/S0895480102412856>

Author's contacts:

Sumalee SANGAMUANG, PhD
Chiang Mai University,
College of Arts, Media and Technology,
239 Huaykaew Rd., Suthep, Muang,
Chiang Mai, 50200 Thailand
sumalee.sa@cmu.ac.th