Partitioning Signed Bipartite Graphs for Classification of Individuals and Organizations

Sujogya Banerjee, Kaushik Sarkar, Sedat Gokalp,
Arunabha Sen, and Hasan Davulcu

Arizona State University
P.O. Box 87-8809, Tempe, AZ, 85281 USA
{sujogya,kaushik.sarkar,sedat.gokalp,asen,hdavulcu}@asu.edu

Abstract. In this paper, we use signed bipartite graphs to model opinions expressed by one type of entities (e.g., individuals, organizations) about another (e.g., political issues, religious beliefs), and based on the strength of that opinion, partition both types of entities into two clusters. The clustering is done in such a way that support for the second type of entity by the first within a cluster is *high* and across the cluster is *low*. We develop an automated partitioning tool that can be used to classify individuals and/or organizations into two disjoint groups based on their beliefs, practices and expressed opinions.

1 Introduction

The goal of the Minerva[1] project, currently underway at Arizona State University is to increase understanding of movements within Muslim communities actively working to counter violent extremism. As a part of this study, we have collected over 800,000 documents from web sites various organizations in Indonesia. Based on the *support* and *opposition* of certain *beliefs* and *practices*, we can partition the set of organizations $\mathcal{O}$ into two groups $\mathcal{O}_1$ and $\mathcal{O}_2$ and the set of beliefs and practices $\mathcal{B}$ into two groups, $\mathcal{B}_1$ and $\mathcal{B}_2$, such that organizations in $\mathcal{O}_1$ support $\mathcal{B}_1$ and oppose $\mathcal{B}_2$, while the organizations $\mathcal{O}_2$ support $\mathcal{B}_2$ and oppose $\mathcal{B}_1$. With the domain knowledge of the social scientists in our team regarding the beliefs and practices of Indonesian community, we can then *label* one group as being *radical* and other as *counter-radical*.

Although the motivation for our work was driven by Minerva, the the problem that is being addressed in this paper is much broader in nature. In the mathematical sociology community, the problem is known as the *Signed two-mode network partitioning problem* [1]. In its mathematical abstraction, the problem is specified by a *bipartite graph* $G = (U \cup V, E)$ and label function $\sigma : E \to \{P, N\}$. The node sets U and V may be representing the set of organizations $\mathcal{O}$ and the set of beliefs $\mathcal{B}$ respectively. If the label of an edge from $o_i \in \mathcal{O}$ to $b_j \in \mathcal{B}$ is P, it implies o_i supports (or has positive opinion) about b_j. If the label of an edge is N, it implies o_i opposes (or has negative opinion) about b_j. The goal of

[1] A project sponsored by the U.S. Department of Defense

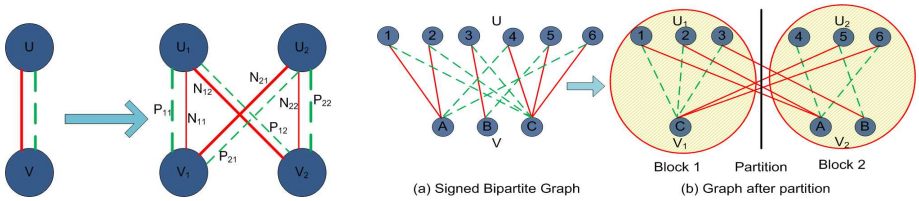


Fig. 1: Partitioning of the node set U and V with the desired goal

the partitioning problem is to divide the node sets U and V into two subsets (U_1, U_2) and (V_1, V_2) respectively, such that

1. number of P edges (positive opinion or support) between nodes within block 1 (P_{11} between U_1 and V_1) and block 2 (P_{22} between U_2 and V_2) is high,
2. number of P edges between nodes across block 1 and block 2 (edges P_{12} between U_1 and V_2 and P_{21} between U_2 and V_1) is low,
3. number of N edges (negative opinion or opposition) between nodes within block 1 (N_{11} between U_1 and V_1) and block 2 (N_{22} between U_2 and V_2) is low and
4. number N edges between nodes across block 1 and block 2 (edges N_{12} between U_1 and V_2 and N_{21} between U_2 and V_1) is high.

The goal of partitioning is depicted in Fig. 1, where the *green* edges indicate support (i.e, P edges) and the *red* edges indicate opposition (i.e, N edges). We can realize these goals by *maximizing* $[(P11 + P22 + N12 + N21) - (P12 + P21 + N11 + N22)]$.

Signed two-mode network partitioning problem can be applied in a multitude of domains, where the node sets U and V can represent different *entities*. For example, (i) U and V may represent the members of the U.S. Senate/House of Representatives and the bills before the senate/house of representatives where they cast their votes, either supporting or opposing the bill; (ii) U and V may represent the political blogs/bloggers and various issues confronting the nation, where they express their opinions either supporting or opposing issues. Clearly, availability of an automated tool that will co-cluster the entities represented by U and V, will be valuable to individuals and organizations that need a coarse grain (two-modal) partitioning of the data set represented by the node set U and V. This tool can help classify individuals or organizations as *radicals vs. counter-radicals*, or *liberals vs. conservatives* or *violent vs. non-violent*, etc.

The main contribution of this effort is the development of a fast automated tool (and associated algorithms) for co-clustering the entities represented by the node sets U and V. We first compute an optimal solution of the partitioning problem using an integer linear program to be used as a benchmark for our heuristic solution. We then develop a heuristic solution and compare its performance using three real data sets. The real data sets include voting records of the Republican and Democratic members of the 111^{th} US Congress and the opinions expressed in top twenty two liberal and conservative blogs. In all these

data sets our partitioning tool produces high quality solution (i.e., with low misclassification) at a low cost (in terms of computation time). To the best of our knowledge, our Minerva research group is the first to present an efficient computational technique for partitioning of signed bipartite graph and apply it to some real data sets.

2 Related works

As the literature on clustering, classification and partitioning is really vast, due to page limitations, we only refer to the ones that are most relevant to this paper [1–4, 8, 7]. The two key features of the partitioning problem addressed in this paper are (i) the graph is *bipartite* and (ii) the weights on the edges are *signed* (i.e., the weights are both positive and negative). Simultaneous clustering of two sets of entities (represented by two sets of nodes in the bipartite graph) was considered in the context of document clustering in [4, 8]. In these studies one set of entities are the documents and the other set is terms or words. Although these efforts study the bipartite graph partition problem, they are distinctly different from our study in one respect. In our study, the edge weights are *signed*, whereas the edges weights considered in [4, 8] are unsigned. Graph partitioning problem with signed edge weights was studied in [2, 3]. However, these studies are also distinctly different from our study in that, while they focus on partitioning general (i.e., arbitrary) graphs, we focus our attention to partitioning bipartite graphs. The study that comes closest to our study is [1, 7], where attention is focused on partitioning of a *signed bipartite* graphs. However, neither [1] nor [7] present any efficient algorithm to solve the partitioning problem in signed bipartite graph.

3 Problem Formulation

In this section we formally define the partitioning problem.

Signed Bipartite Graph Partition Problem (SBGPP): An edge labeled weighted bipartite graph $G = (U \cup V, E)$ where $U = \{u_1, u_2, \ldots, u_n\}$ represents entities of type I and $V = \{v_1, v_2, \ldots, v_m\}$ represents entities of type II. Each edge $(u, v) \in E$ has two functions associated with it: (i) label function $\sigma : E \to \{P, N\}$, which indicates the type of opinion (positive or negative), and (ii) weight function $w : E \to \mathbb{Z}$, which indicates the strength of that opinion. $A_N = [w_n(u, v)]$ and $A_P = [w_p(u, v)]$ are the weighted adjacency matrix for edges with label N and P respectively. If the node set U is partitioned into U_1 and U_2 and V is partitioned into V_1 and V_2, the strength of the positive and negative opinions of the entities of type I regarding the entities of type II are defined as follows:
For all edges $(u, v) \in E$,

$$P_{11} = \sum_{u \in U_1} \sum_{v \in V_1} w_p(u, v),\ P_{12} = \sum_{u \in U_1} \sum_{v \in V_2} w_p(u, v),\ P_{22} = \sum_{u \in U_2} \sum_{v \in V_2} w_p(u, v)$$

$$P_{21} = \sum_{u \in U_2} \sum_{v \in V_1} w_p(u, v),\ N_{11} = \sum_{u \in U_1} \sum_{v \in V_1} w_n(u, v),\ N_{12} = \sum_{u \in U_1} \sum_{v \in V_2} w_n(u, v)$$

$$N_{22} = \sum_{u \in U_2} \sum_{v \in V_2} w_n(u,v),\ N_{21} = \sum_{u \in U_2} \sum_{v \in V_1} w_n(u,v)$$

Problem: Find a partition of the node set U into U_1 and U_2 and V into V_1 and V_2 such that $[(P11 + P22 + N12 + N21) - (P12 + P21 + N11 + N22)]$ is *maximized*.

4 Computational Techniques

In this section we give a mathematical programming technique to find the optimal solution for the SBGPP. Since computational time for finding optimal solution for large graphs is unacceptably high, we present a heuristic in subsequent section to solve the SBGPP.

4.1 Optimal Solution for SBGPP

The goal of the SBGPP is to partition U into two disjoint sets U_1 and U_2 (similarly V into V_1 and V_2). For each node in $u \in U$ and each partition $U_i,\ i = 1, 2$, we use a variable b_{ui}. b_{ui} is 1 iff in u is in U_i. Similarly we define variable p_{vi} for all $v \in V$. We will refer $B_1 = U_1 \cup V_1$ and $B_2 = U_2 \cup V_2$ as *blocks* 1 and 2 respectively.

Variables: For each node $u \in U,\ v \in V$ and each partition $U_i,\ V_i,\ i = 1, 2$

$$b_{ui} = \begin{cases} 1, & \text{if node } u \text{ is in partition } U_i \\ 0, & \text{otherwise.} \end{cases} \qquad p_{vi} = \begin{cases} 1, & \text{if node } v \text{ is in partition } V_i \\ 0, & \text{otherwise.} \end{cases}$$

The mathematical programming formulation is given as follows:

$$\begin{aligned} \max \quad L = & \sum_{i=1}^{2} \sum_{u \in U_i} \sum_{v \in V_i} (w_p(u,v) - w_n(u,v)) b_{ui} p_{vi} \\ & + \sum_{\substack{i,j=1 \\ i \neq j}}^{2} \sum_{u \in U_i} \sum_{v \in V_j} (w_n(u,v) - w_p(u,v)) b_{ui} p_{vj} \\ s.t \quad & b_{u1} + b_{u2} = 1, \quad \forall u \in U \qquad (1) \\ & p_{u1} + p_{u2} = 1, \quad \forall p \in V \qquad (2) \end{aligned}$$

The objective function computes the objective value given by the expression L. We want to maximize L. It may be noted that the above quadratic objective function can easily be changed into a linear function by simple variable transformation [6]. Constraint 1 and 2 ensures that each node in U and V belongs to one particular block.

4.2 Move-based Heuristics

We present a move-based heuristic to find an approximate solution of SBGPP. The move-based heuristic is a variant of well known FM algorithm [5] for partitioning graphs. The algorithm starts with a random initial partition and iteratively moves nodes from one block to another such that the value of the objective

function is improved. The *"gain"* of a node is defined as the value by which the objective function increases if the node is moved from one block to the other. In each iteration the node with the highest gain is moved from one block to the other. In case of a tie a node is chosen arbitrarily. After a node is moved, it is locked and is not moved until the next pass. The heuristic is presented in Algorithm 1. It should be noted that original FM algorithm will not work for our problem as SBGPP relates to signed bipartite graphs with a completely different objective function and doesn't have any size constraints. As a result the node gain computation routine Algorithm 2 is considerably different from the original FM algorithm. Algorithm 1 runs for r different initial random partition of the nodes to avoid the possibility of being stuck at a local maxima. In practice the heuristic converges very fast, mostly in 2 to 3 passes.

Algorithm 1: Move-based Heuristic (MBH)

Input : A weighted signed bipartite graph $H = (U \cup V, E)$
Output: A partition of the nodes U_1, U_2 and V_1, V_2 such that objective value L is maximum

```
1  L ⟵ 0;
2  for i ⟵ 1 to r do
3      Generate a random partitioning of the nodes in U into U1 and U2 and nodes
       in V into V1 and V2;
4      repeat
5          Compute gains of all nodes using Algorithm 2 ;
6          repeat
7              Among all the unlocked nodes select the node of highest gain. Move
               the node to the other block and call it base node. Lock the base
               node;
8              Update the node gains of all the free neighbors of the base node;
9          until Until all the nodes are locked;
10         Change the current partition into a new partition that has the largest
           value of the objective function in this pass ;
11         Unlock all the nodes;
12     until If the objective value L' improves during the last pass;
13     if L' ≥ L then L' ← L and save the current partition
14 return L and the final partition of nodes
```

5 Experimental Results and Discussions

To validate the effectiveness of our heuristic and benchmark its performance we tested the heuristic both on synthetic and real world data. The real world data consists of *US Congress* (SENATE, REP) and *political blogosphere* (BLOG) data sets.

5.1 US Congress Data [SENATE, REP]

The US Congress has been collecting data since the very first congress of the US history. This data has been encoded as XML files and publicly shared through

Algorithm 2: Node Gain Computation

Input : A weighted signed bipartite graph $G = (U \cup V, E)$
Output: Gains of all nodes

```
1 foreach node u ∈ U ∪ V do
2     gain(u) ⟵ 0;
      // FBlock = "from block" of node u, ToBlock = "to block" of node
         u, w(e) = weight of edge e and # = number
3     foreach edge e ∈ E with l(e) = N of node u do
4         if # nodes of e in ToBlock is 0 then gain(u) ← gain(u) + 2 * w(e);
5         if # nodes of e in FBlock is 1 then gain(u) ← gain(u) − 2 * w(e);
6     foreach edge e ∈ E with l(e) = P of node u do
7         if # nodes of e in ToBlock is 0 then gain(u) ← gain(u) − 2 * w(e);
8         if # nodes of e in FBlock is 1 then gain(u) ← gain(u) + 2 * w(e);
```

the govtrack.us project[2]. From various types of data available at the project site, we collected the *roll call votes* for the 111$^{\text{th}}$ US Congress which includes The Senate and The House of Representatives and covers the years 2009-2010. The 111$^{\text{th}}$ Senate data contains information about 108 senators and their votes on 696 bills[3]. The 111$^{\text{th}}$ Congress has 451 representatives and the data contains their vote on 1655 bills.

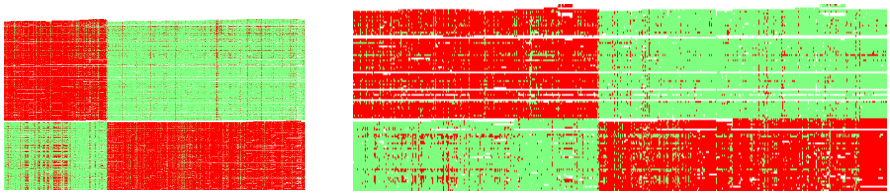
We extracted the SENATE and REP data in adjacency matrices $A_{|U|\times|V|}$, with U vertices representing the congressmen, and the V vertices representing the bills. The edge (u_i, v_j), $u_i \in U, v_j \in V$ has weight 1 if the congressman u_i votes 'Yea' for the bill v_j , -1 if the congressman votes 'Nay', and 0 if he did not attend the session. We have the original classification vector for both the congressmen and the bills in terms of which party they represent (or which party sponsored the bill). The first two columns of Table 1 provide information about this data as well as the partitioning accuracies of the algorithms. Figure 2 depicts the partitioned vote matrices of the 111$^{\text{th}}$ US Congress data, where rows representing the congressmen and the columns representing the bills. Also, the light green color represents 'Yea' votes, and dark red represents 'Nay' votes.

5.2 Blog Data [BLOG]

As Web 2.0 platforms gained popularity, it became easy for web users to be a part of the web and express their opinions, mostly through blogs. Most blogs are maintained by individuals, whereas there are also professional blogs with a group of authors. In this study, we focus on a set of popular political liberal or conservative blogs that have a clearly declared positions. These blogs contain discussions about social, political, economic issues and related key individuals.

[2] http://www.govtrack.us/data

[3] Normally, each congress has 100 senators (2 from each state), however in many of the congresses, there are unexpected changes on the seats caused by displacements or deaths.



(a) 111th US House (b) 111th US Senate

Fig. 2: Vote matrix of US Congress after partitioning

Table 1: Descriptive summaries of the graphs for each dataset with the Heuristic accuracy

	111th US Senate	111th US House	Political Blogosphere
Vertices in U	64 Democrat 42 Republican Senator	268 Democrat 183 Republican Representatives	13 Liberal 9 Conservative Blogs
Vertices in V	696 Bills	1655 Bills	20 Liberal 14 Conservative People
Graph Density	88.36 %	91.23 %	39.04 %
Heuristic accuracy	100.00%	99.56%	98.21%

They express positive sentiment towards individuals whom they share ideologies with, and negative sentiment towards the others. In these blogs, it is also common to see criticism of people within the same camp, and also support for people from the other camp.

In this experiment, we collected a list of 22 most popular liberal and conservative blogs from the Technorati[4] rankings. For each blog, we fetched the posts for the period of 6 months before the 2008 US presidential elections (May - October, 2008). We expected to have high intensity of the debates and discussions and resulting in a bipolar clustering in the data. Table 2 shows the partial list of blogs with their URLs, political camps and the number of posts for the given period.

We use AlchemyAPI[5] to run a named entity tagger to extract the people names mentioned in the posts, and an entity-level sentiment analysis which provided us with weighted and signed sentiment (positive values indicating support, and negative indicating opposition) for each person. This information was used to synthesize a signed bipartite graph (the BLOG data), where the blogs and people correspond to the two sets of vertices U and V. The a_{ij} values of the adjacency matrix A are the cumulative sum of sentiment values for each mention of the person v_j by the blog u_i.

[4] http://technorati.com
[5] http://www.alchemyapi.com

To get a gold standard list of the most influential liberal and conservative people, we used The Telegraph List[6] for 2007. The third column of Table 1 provides information about this data as well as the partitioning accuracies of the algorithm.

Table 2: Political Blogs

Blog name	URL	Political view	Size
Huffington Post	http://www.huffingtonpost.com/	Liberal	3959
Daily Kos	http://www.dailykos.com/	Liberal	1957
Boing Boing	http://www.boingboing.net/	Liberal	1576
Crooks and Liars	http://www.crooksandliars.com/	Liberal	1497
Firedoglake	http://www.firedoglake.com/	Liberal	1354
Hot Air	http://hotair.com/	Conservative	1579
Reason - Hit and Run	http://reason.com/blog	Conservative	1563
Little green footballs	http://littlegreenfootballs.com/	Conservative	787
Atlas shrugs	http://atlasshrugs2000.typepad.com/	Conservative	773
Stop the ACLU	http://www.stoptheaclu.com/	Conservative	741
Wizbangblog	http://wizbangblog.com/	Conservative	621

6 Conclusion

In this paper we study the problem of partitioning *signed bipartite graph* with relevant application in political, religious and social domains. We provided a fast heuristic to find the solution for this problem. We tested the high accuracy of our heuristic on three sets of real data collected from political domain.

References

1. Andrej, M., Doreian, P.: Partitioning signed two-mode networks. Journal of Mathematical Sociology 33, 196–221 (2009)
2. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. In: MACHINE LEARNING. pp. 238–247 (2002)
3. Charikar, M., Guruswami, V., Wirth, A.: Clustering with qualitative information. In: Proceedings of the 44th Annual IEEE FOCS (2003)
4. Dhillon, I.S.: Co-clustering documents and word using bipartite spectral graph partitioning. In: Proceedings of the KDD. IEEE (2001)
5. Fiduccia, C., Mattheyses, R.: A linear-time heuristic for improving network partitions. In: Papers on Twenty-five years of electronic design automation. pp. 241–247. ACM (1988)
6. Sen, A., Deng, H., Guha, S.: On a graph partition problem with application to vlsi layout. Inf. Process. Lett. 43(2), 87–94 (1992)
7. Zaslavsky, T.: Frustration vs. clusterability in two-mode signed networks (signed bipartite graphs) (2010)
8. Zha, H., He, X., Ding, C., Simon, H., Gu, M.: Bipartite graph partitioning and data clustering. In: Proceedings of the 10th International Conference on Information and Knowledge Management. pp. 25–32. ACM (2001)

[6] The-top-US-conservatives-and-liberals.html