Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media

Quantifying Political Polarity Based on Bipartite Opinion Networks

Leman Akoglu
Department of Computer Science
Stony Brook University, Stony Brook, NY, 11794
leman@cs.stonybrook.edu

Abstract

Political inclinations of individuals (liberal vs. conservative) largely shape their opinions on several issues such as abortion, gun control, nuclear power, etc. These opinions are openly exerted in online forums, news sites, the parliament, and so on. In this paper, we address the problem of quantifying political polarity of individuals and of political issues for classification and ranking. We use signed bipartite networks to represent the opinions of individuals on issues, and formulate the problem as a node classification task. We propose a linear algorithm that exploits network effects to learn both the polarity labels as well as the rankings of people and issues in a completely unsupervised manner. Through extensive experiments we demonstrate that our proposed method provides an effective, fast, and easy-to-implement solution, while outperforming three existing baseline algorithms adapted to signed networks, on real political forum and US Congress datasets. Experiments on a wide variety of synthetic graphs with varying polarity and degree distributions of the nodes further demonstrate the robustness of our approach.

Introduction

Many individuals use online media to exert their opinions on a variety of topics. Hotly debated topics include liberal vs. conservative policies such as tax cuts and gun control, social issues such as abortion and same-sex marriage, environmental issues such as climate change and nuclear power plants, etc. These openly debated issues in blogs, forums, and news websites shape the nature of public opinion and affect the direction of politics, media, and public policy.

In this paper, we address the problem of quantifying political polarity in opinion datasets. Given a set of individuals from two opposing camps (liberal vs. conservative) debating a set of issues or exerting opinions on a set of subjects (e.g. human subjects, political issues, congressional bills), we aim to address two problems: (1) *classify* which person lies in which camp, and which subjects are favored by each camp; and (2) *rank* the people and the subjects by the magnitude or extent of their polarity. Here while the classification enables us to determine the two camps, ranking helps us understand the extremity to which a person/subject is polarized; e.g. same-sex marriage may be highly polarized among the two camps (liberals being strictly in favor, and conservatives strictly being against), while gun control may not belong to a camp fully favoring or opposing it (i.e., is less polarized). Ranking also helps differentiate moderate vs. extreme individuals, as well as unifying vs. polarizing subjects; such as (unifying) bills voted in the same way, e.g. all 'yea' by the majority of congressmen vs. (polarizing) bills that are voted quite oppositely by the two camps.

A large body of prior research focuses on sentiment analysis on politically oriented text (Cohen and Ruths 2013; Conover et al. 2011b; 2011a; Pak and Paroubek 2010; Tumasjan et al. 2010). The main goal of these works is to classify political text. In this work, on the other hand, we deal with network data to classify its nodes. Moreover, these methods mostly rely on supervised techniques whereas we focus on un/semi-supervised classification. Other prior research on polarization have exploited link mining and graph clustering to study the social structure on social media networks (Adamic and Glance 2005; Livne et al. 2011; Conover et al. 2011b; Guerra et al. 2013) where the edges depict the 'mention' or 'hyperlink' relations and not opinions. Moreover, these works do not perform ranking.

Different from previous works, our key approach is to exploit network effects to both classify and rank individuals and political subjects by their polarity. The opinion datasets can be effectively represented as signed bipartite networks, where one set of nodes represent individuals, the other set represent subjects, and signed edges between the individuals and subjects depict the $+/-$ opinions. As such, we cast the problem as a node classification task on such networks.

Our main contributions can be summarized as follows:

- We cast the political polarity classification and ranking problem as a node classification task on edge-signed bipartite opinion networks.
- We propose an algorithm, called signed polarity propagation (sPP), that computes probabilities (i.e., polarity scores) of nodes of belonging to one of two classes (e.g., liberal vs. conservative), and use these scores for classification and ranking. Our method is easy-to-implement and fast—running time grows linearly with network size.
- We show the effectiveness of our algorithm, in terms of both prediction and ranking, on synthetic and real datasets with ground truth from the US Congress and Political Forum. Further, we modify three existing algorithms to

handle signed networks, and compare them to sPP. Our experiments reveal the advantages and robustness of our method on diverse settings with various polarity and degree distributions.

Related Work

Scoring and ranking the nodes of a graph based on the network structure has been studied extensively, with well-known algorithms like PageRank (Brin and Page 1998), and HITS (Kleinberg 1998). These, however, are applied on graphs where the edges are unsigned and therefore cannot be directly used to compute polarity scores.

Computing polarity scores can be cast as a network classification problem, where the task is to assign probabilities (i.e., scores) to nodes of belonging to one of two classes, which is the main approach we take in our work. There exist a large body of work on network-based classification (Getoor et al. 2007; Neville and Jensen 2003; Macskassy and Provost 2003). Semi-supervised algorithms based on network propagation have also been used in classifying political orientation (Lin and Cohen 2008; Zhou, Resnick, and Mei 2011). However, all the existing methods work with unsigned graphs in which the edges do not represent opinions but simply relational connections such as HTML hyperlinks between blog articles or 'mention' relations in Twitter.

Signed networks have only recently attracted attention (Leskovec, Huttenlocher, and Kleinberg 2010b). Most existing studies focused on tackling the edge sign prediction problem (Yang et al. 2012; Chiang et al. 2011; Leskovec, Huttenlocher, and Kleinberg 2010a; Symeonidis, Tiakas, and Manolopoulos 2010). Other works include the study of trust/distrust propagation (Guha et al. 2004; DuBois, Golbeck, and Srinivasan 2011; Huang et al. 2012), and product/merchant quality estimation from reviews (McGlohon, Glance, and Reiter 2010). These works do not address the problem of classifying and ranking nodes in signed graphs.

With respect to studies on political orientation through social media, (Adamic and Glance 2005; Adar et al. 2004) use link mining and graph clustering to analyze political communities in the blogosphere. While these and most clustering algorithms are designed to work with unsigned graphs, there also exist approaches for clustering signed graphs (Traag and Bruggeman 2009; Lo et al. 2013; Zhang et al. 2013). Clustering, however, falls short in scoring the nodes and hence quantifying polarity for ranking.

Related, (Livne et al. 2011) utilize graph and text mining techniques to analyze differences between political parties and their online media usage in conveying a cohesive message. Most recently, (Cohen and Ruths 2013) use supervised classification techniques to classify three groups of Twitter users with varying political activity (figures, active, and modest) by their political orientation. Other works that exploit supervised classification using text features include (Conover et al. 2011a; Golbeck and Hansen 2011; Pennacchiotti and Popescu 2011).

Similar to (Adamic and Glance 2005), there exist related works that study the social structure for measuring polarity. These works rely on the existence of (unsigned) social links between the users of social media and study the communities induced by polarized debate; an immediate consequence of the homophily principle, which states that people with similar beliefs and opinions tend to establish social ties. (Livne et al. 2011) and (Conover et al. 2011b) both use modularity (Newman 2006) as a measure of segregation between political groups in Twitter. (Guerra et al. 2013) compare modularity of polarized and non-polarized networks and propose two new measures of polarity based on community boundaries. Again, these works do not study the opinion networks with signed edges.

As our main task is quantifying polarity, work on sentiment analysis is also related. There exist a long list of works on sentiment and polarity prediction in political text (Tumasjan et al. 2010; Awadallah, Ramanath, and Weikum 2010; Conover et al. 2011b; He et al. 2012), as well as in tweets, blogs, and news articles (Pak and Paroubek 2010; Godbole, Srinivasaiah, and Skiena 2007; Thomas, Pang, and Lee 2006; Balasubramanyan et al. 2012). These differ from our work as they use text-based sentiment analysis, while we focus on the network effects in signed graphs.

Proposed Method

Problem Overview

We consider the problem of polarity prediction and ranking in opinion datasets (e.g. forums, blogs, the congress). Opinion datasets mainly consist of a set of people (e.g. users in a forum, representatives in The House) and a set of subjects (e.g. political issues, political people, congressional bills). Each person often maintains a positive or negative opinion toward a particular subject (e.g. a representative votes 'yes' for a bill, a person 'likes' a political individual). This opinion is often an exposition of the person's latent political leaning—liberal or conservative. For example, we could think of a person with strong negative opinion toward gay&lesbian rights to be more conservative. As such, the subjects can also be grouped into several classes—liberal- or conservative-favored.

Our goal is twofold. First we aim to *predict* the latent political *classes* of the people as well as the subjects in opinion datasets. Second, we aim to *rank* both the people and the subjects with respect to the *magnitude* of their polarity in the political spectrum.

An opinion dataset can be represented as a bipartite graph with signed edges in which person nodes are connected to subject nodes, with links representing the $+/-$ opinions. The objects in this network belong to certain classes (e.g. liberal and conservative). As such, we tackle the above problems by formulating a graph-based classification objective, estimate the class probabilities for prediction and furthermore use the estimated probabilities for ranking.

Problem Formulation

Notation We are given a signed bipartite opinion network $G_s = (N, E)$, in which a set of n person nodes $\mathcal{U} = \{u_1, \ldots, u_n\}$ and a set of m subject nodes $\mathcal{V} = \{v_1, \ldots, v_m\}$ are connected with signed edges $e(u_i, v_j, s) \in$

E, where edge signs $s \in \{+,-\}$ depict positive and negative opinions of people toward subjects, and $\mathcal{U} \cup \mathcal{V} = N$. A neighborhood function $\mathcal{N}$, $\mathcal{N}_{u_i} \subseteq \mathcal{V}$ and $\mathcal{N}_{p_j} \subseteq \mathcal{U}$, describes the underlying bipartite network structure.

In our proceeding formulation, each node in N is represented as a random variable that takes a value from an appropriate class label domain; in our case, $\mathcal{L}_{\mathcal{U}} = \{L, C\}$ (L for $liberal$ and C for $conservative$), and $\mathcal{L}_{\mathcal{V}} = \{LF, CF\}$ (LF for $liberal\text{-}favored$ and CF for $conservative\text{-}favored$). In this classification task, we denote by $\mathcal{Y}^N = \mathcal{Y}^{\mathcal{U}} \cup \mathcal{Y}^{\mathcal{V}}$ the nodes the labels of which need to be assigned, and let y_i refer to Y_i's label.

Objective formulation Next we define the objective function for our classification task. The key idea is to consider this task as network-based classification. We propose to use an objective formulation that utilizes pairwise Markov Random Fields (MRFs) (Kindermann and Snell 1980), which we adapt to our problem setting.

MRFs are a class of probabilistic graphical models that are suited for solving inference problems in networked data. An MRF consists of an undirected graph where each node can be in any of a finite number of states (in our case, class labels). The state of a node is assumed to be dependent on each of its neighbors and independent of other nodes in the graph.[1] In pairwise MRFs, the joint probability of the graph can be written as a product of pairwise factors, parameterized over the edges. The factors are also referred as clique potentials in general MRFs, which are essentially functions that collectively determine the graph's joint probability.

Specifically, let $G_s = (N, E)$ denote a signed network of random variables as before, where N consists of the unobserved variables $\mathcal{Y}$ which need to be assigned values from the label set $\mathcal{L} = \mathcal{L}_{\mathcal{U}} \cup \mathcal{L}_{\mathcal{V}}$. Let Ψ denote a set of clique potentials that consists of two types of factors:

- For each $Y_i \in \mathcal{Y}^{\mathcal{U}}$ and $Y_j \in \mathcal{Y}^{\mathcal{V}}$, $\psi_i, \psi_j \in \Psi$ are *prior* mappings $\psi_i^{\mathcal{U}} : \mathcal{L}_{\mathcal{U}} \rightarrow \mathbb{R}_{\geq 0}$, and $\psi_j^{\mathcal{V}} : \mathcal{L}_{\mathcal{V}} \rightarrow \mathbb{R}_{\geq 0}$, where $\mathbb{R}_{\geq 0}$ denotes non-negative real numbers.
- For each $e(Y_i^{\mathcal{U}}, Y_j^{\mathcal{V}}, s) \in E$, $\psi_{ij}^s \in \Psi$ is a *compatibility* mapping $\psi_{ij}^s : \mathcal{L}_{\mathcal{U}} \times \mathcal{L}_{\mathcal{V}} \rightarrow \mathbb{R}_{\geq 0}$.

Given an assignment $\mathbf{y}$ to all the unobserved variables $\mathcal{Y}^N$ and $\mathbf{x}$ to observed ones $\mathcal{X}^N$ (variables with known values, if any), our objective function is associated with the joint probability distribution

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{Y_i \in \mathcal{Y}^N} \psi_i(y_i) \prod_{e(Y_i^{\mathcal{U}}, Y_j^{\mathcal{V}}, s) \in E} \psi_{ij}^s(y_i, y_j) \quad (1)$$

where $Z(\mathbf{x})$ is the normalization function. Our goal is to infer the maximum likelihood assignment of states (labels) to unobserved variables (nodes) that will maximize the objective function above.

Problem definition Having introduced the network classification task and our objective, we define the polarity classification and ranking problems formally.

Given

- a bipartite opinion network $G_s = (N, E)$ of people and subjects connected with *signed* edges,
- *compatibility* of two objects with a given pair of labels being connected to each other, and
- *prior* knowledge (probabilities) of network objects belonging to each class;

P1. Classify the network objects $Y_i \in \mathcal{Y}^N$, into one of two respective classes; $\mathcal{L}_{\mathcal{U}} = \{L, C\}$, and $\mathcal{L}_{\mathcal{V}} = \{LF, CF\}$, where the class assignments y_i maximize Equation (1), and
P2. Rank the network objects by the magnitude of their political polarity.

Signed Polarity Propagation Algorithm

Finding the best assignments to unobserved variables in our objective function, i.e. **P1** above, is the inference problem. The brute force approach through enumeration of all possible assignments is exponential and thus intractable. In general, exact inference is known to be NP-hard and there is no known algorithm which can be theoretically shown to solve the inference problem for general MRFs. Therefore, we employ a computationally tractable (in fact linearly scalable with network size) approximate inference algorithm called Loopy Belief Propagation (LBP) (Yedidia, Freeman, and Weiss 2003), on signed political networks.

The SPP algorithm applied on a *signed* bipartite network is based on iterative message passing and can be concisely expressed as the following equations:

$$m_{i \rightarrow j}(y_j) = \alpha_1 \sum_{y_i \in \mathcal{L}^{\mathcal{U}}} \psi_{ij}^s(y_i, y_j)\, \psi_i^{\mathcal{U}}(y_i) \prod_{Y_k \in \mathcal{N}_i \cap \mathcal{Y}^{\mathcal{V}} \setminus Y_j} m_{k \rightarrow i}(y_i), \;\; \forall y_j \in \mathcal{L}^{\mathcal{V}} \quad (2)$$

$$b_i(y_i) = \alpha_2\, \psi_i^{\mathcal{U}}(y_i) \prod_{Y_j \in \mathcal{N}_i \cap \mathcal{Y}^{\mathcal{V}}} m_{j \rightarrow i}(y_i), \forall y_i \in \mathcal{L}^{\mathcal{U}} \quad (3)$$

where $m_{i \rightarrow j}$ is a *message* sent by person i to subject j (a similar equation can be written for messages from subjects to people), and it captures the belief of i about j, which is the probability distribution over the labels of j, i.e. what i 'thinks' j's label is, given the current label of i and the sign of the edge that connects i and j. Beliefs refer to marginal probability distributions of nodes over labels; for example $b_i(y_i)$ denotes the *belief* of person i having label y_i (again, a similar equation can be written for beliefs of subjects). α's are the normalization constants, which respectively ensure that each message and each set of marginal probabilities sum to 1. At every iteration, each node computes its belief based on messages received from its neighbors, and uses the compatibility mapping to transform its belief into messages for its neighbors. The key idea is that after enough iterations of message passes between the nodes, the 'conversations' are likely to come to a consensus, which determines the marginal probabilities of all the unknown variables.

The details of the SPP algorithm is given in Algorithm 1, which first initializes the messages and the *prior*s (lines 3-6). It then proceeds by making each set of $Y_i \in \mathcal{Y}^{\mathcal{U}}$ and $Y_j \in \mathcal{Y}^{\mathcal{V}}$ alternately communicate messages with their neighbors in an iterative fashion as discussed before until the

[1] This assumption yields a pairwise MRF which is a special case of general MRFs (Yedidia, Freeman, and Weiss 2003).

Algorithm 1: SIGNED POLARITY PROPAGATION

1 **Input:** Bipartite graph G_s=(N, E) of people, subjects, and $+/-$ opinions, compatibility potentials ψ_{ij}^s
2 **Output:** Polarity label for every person i and subject j
3 **foreach** $e(Y_i, Y_j, s) \in E$ *s.t.* $Y_i, Y_j \in \mathcal{Y}^N$ **do** // *initialize*
4   **foreach** $y_i \in \mathcal{L}^{\mathcal{U}}$, $y_j \in \mathcal{L}^{\mathcal{V}}$ **do**
5         $m_{i \to j}(y_j) \leftarrow 1,\ \ \phi_i^{\mathcal{U}}(y_i) \leftarrow 1/|\mathcal{L}_{\mathcal{U}}|$
6         $m_{j \to i}(y_i) \leftarrow 1,\ \ \phi_j^{\mathcal{V}}(y_j) \leftarrow 1/|\mathcal{L}_{\mathcal{V}}|$
7 **repeat**// *perform message propagation*
8   // *update messages from people to subjects*
9   **foreach** $e(Y_i, Y_j, s) \in E$ *s.t.* $Y_i, Y_j \in \mathcal{Y}^N$ **do**
10   **foreach** $y_j \in \mathcal{L}^{\mathcal{V}}$ **do**
11             $m_{i \to j}(y_j) \leftarrow \alpha_1 \sum_{y_i \in \mathcal{L}^{\mathcal{U}}} \psi_{ij}^s(y_i, y_j) \phi_i^{\mathcal{U}}(y_i) \prod_{Y_k \in \mathcal{N}_i \cap \mathcal{Y}^{\mathcal{V}} \setminus Y_j} m_{k \to i}(y_i)$
12   // *update messages from subjects to people*
13   **foreach** $e(Y_i, Y_j, s) \in E$ *s.t.* $Y_i, Y_j \in \mathcal{Y}^N$ **do**
14   **foreach** $y_i \in \mathcal{L}^{\mathcal{U}}$ **do**
15             $m_{j \to i}(y_i) \leftarrow \alpha_3 \sum_{y_j \in \mathcal{L}^{\mathcal{V}}} \psi_{ij}^s(y_i, y_j) \phi_j^{\mathcal{V}}(y_j) \prod_{Y_k \in \mathcal{N}_j \cap \mathcal{Y}^{\mathcal{U}} \setminus Y_i} m_{k \to j}(y_j)$
16 **until** all messages $m(y)$ stop changing
17 **foreach** $Y_i, Y_j \in \mathcal{Y}^N$ **do** // *compute beliefs*
18   **foreach** $y_i \in \mathcal{L}^{\mathcal{U}}$, $y_j \in \mathcal{L}^{\mathcal{V}}$ **do**
19         $b_i(y_i) = \alpha_2 \phi_i^{\mathcal{U}}(y_i) \prod_{Y_j \in \mathcal{N}_i \cap \mathcal{Y}^{\mathcal{V}}} m_{j \to i}(y_i)$
20         $b_j(y_j) = \alpha_4 \phi_j^{\mathcal{V}}(y_j) \prod_{Y_i \in \mathcal{N}_j \cap \mathcal{Y}^{\mathcal{U}}} m_{i \to j}(y_j)$

messages stabilize (lines 7-16), i.e. convergence is reached.[2] At convergence, we calculate the marginal probabilities, that is of assigning Y_i with label y_i, by computing the final belief $b_i(y_i)$ (lines 16-20).

We use these maximum likelihood label probabilities for classification; e.g. for each person u_i, we assign the label $\mathcal{L}_{u_i} \leftarrow max_{y_i} b_i(y_i)$. As for ranking, i.e. **P2** above, we sort the objects by class probabilities, e.g. $r^{\mathcal{U}}_{\text{SPP}} \leftarrow sort(b_i(y_i = L))$ sorts people from most conservative to most liberal.

To completely define the SPP algorithm, we need to instantiate the clique potential factors Ψ, i.e. compatibilities and priors, which we discuss next.

Priors. The *prior* beliefs $\psi_i^{\mathcal{U}}$ and $\psi_j^{\mathcal{V}}$, respectively of people and subjects, can be suitably initialized if there is any prior knowledge of the objects (e.g. Barack Obama is a $liberal$ person). As such, our method is flexible to integrate available side information. In case there is no prior knowledge available, each node is initialized equally likely to have any of the possible labels, i.e. $\frac{1}{|\mathcal{L}|}$ as in Algorithm 1 (lines 5-6).

Compatibility matrices. The *compatibility* potentials can be thought of as matrices with entries $\psi_{ij}^s(y_i, y_j)$, which gives the likelihood of a node having label y_i, given that it has a neighbor with label y_j on which s/he exhibits a $s \in \{+, -\}$ opinion, i.e. the edge sign is s. Note that unlike earlier models, we use clique potentials based on the edge labels s in our formulation. This is exactly because the compatibility of class labels of two adjacent nodes depends on the sign of the edge connecting them: e.g., $L \xrightarrow{+} LF$ is highly compatible, whereas $L \xrightarrow{-} LF$ is not likely. A sample instantiation of compatibilities is shown in Table 1.

Note that while LBP has been theoretically studied for uniqueness on signed graphs (Watanabe 2011), we are the first to formulate and employ signed LBP for the political polarity problems **P1** and **P2** in practice.

[2] Although convergence is not theoretically guaranteed, in practice LBP converges to beliefs within a small threshold of change (e.g. $\epsilon = 10^{-4}$) quickly with accurate results (Pandit et al. 2007).

Table 1: Instantiation of opinion-based compatibility potentials. Entry $\psi_{ij}^s(y_i, y_j)$ is the compatibility of a subject node having label y_j having a person node neighbor with label y_i, given the opinion of i on j is $s \in +/-$, for small ϵ.

s: '+'	**Subjects**	
People	LF	CF
L	1-ϵ	ϵ
C	ϵ	1-ϵ

s: '$-$'	**Subjects**	
People	LF	CF
L	ϵ	1-ϵ
C	1-ϵ	ϵ

Competitor Methods

In this section we describe three alternative methods that we compare to in the experiments. In particular, we modify three well-known clustering/classification algorithms to handle signed networks for our problem setting: weighted-vote relational classifier (Macskassy and Provost 2003), hubs-and-authorities (HITS) algorithm (Kleinberg 1998), and spectral clustering (Ng, Jordan, and Weiss 2001).

weighted-vote Relational Network Classifier

The *wvRN* is a neighbor-based classifier by Mackassy and Provost (Macskassy and Provost 2003), which estimates class-membership probability of each node as the weighted mean of the class-membership probabilities of its neighbors. In our setting the underlying network is bipartite and the edges are signed, thus the above definition translates to:

$$\Pr_i^{\mathcal{U}}(L) = \frac{1}{Z_i}\Bigg(\sum_{j \in \mathcal{N}_i^+} w_{ij}^+ \Pr_j^{\mathcal{V}}(LF) - \sum_{j \in \mathcal{N}_i^-} w_{ij}^- \Pr_j^{\mathcal{V}}(CF)\Bigg)$$

$$\Pr_j^{\mathcal{V}}(LF) = \frac{1}{Z_j}\Bigg(\sum_{i \in \mathcal{N}_j^+} w_{ij}^+ \Pr_i^{\mathcal{U}}(L) - \sum_{i \in \mathcal{N}_j^-} w_{ij}^- \Pr_i^{\mathcal{U}}(C)\Bigg)$$

where $\mathcal{N}^+$ ($\mathcal{N}^-$) denotes the neighbors of a node that are linked to it by positive (negative) weights w^+ (w^-). In our case, w^+=1 and w^-=-1, for positive and negative opinions. Z's are normalization constants, i.e. $Z_i = \sum_{j \in \mathcal{N}_i^+} w_{ij}^+ - \sum_{j \in \mathcal{N}_i^-} w_{ij}^- = |\mathcal{N}_i|$. Finally, $\Pr_i^{\mathcal{U}}(C) = 1 - \Pr_i^{\mathcal{U}}(L)$ and $\Pr_j^{\mathcal{V}}(CF) = 1 - \Pr_j^{\mathcal{V}}(LF)$; $\Pr_i^{\mathcal{U}}, \Pr_j^{\mathcal{V}} \in [0, 1]$.

We use the above equations to iteratively update class probabilities of all nodes. Nodes with unknown labels are initially assigned class priors. Due to the loopy nature of propagation, convergence is not guaranteed, although in our experiments the probabilities converged within a small threshold of change (ϵ=10^{-4}) in consecutive iterations.

The class-membership probabilities are used for classification; e.g. for people $\mathcal{L}_{u_i} \leftarrow max_{y_i}\mathrm{Pr}_i^{\mathcal{U}}(y_i)$. For ranking we sort each set of the network objects by one of the class probabilities; e.g. for people $r^{\mathcal{U}}_{wvRN} \leftarrow sort(\mathrm{Pr}_i^{\mathcal{U}}(y_i = L))$.

Signed HITS Algorithm

We also adapt Kleinberg's HITS algorithm (Kleinberg 1998) to compute the *liberality* of the people and the subjects. These values are defined in terms of one another in a mutual recursion: the liberality of a person/subject is the scaled sum of the liberality values of subjects/people linked to it by positive opinion minus the sum of those linked by negative opinion. We give the corresponding equations below.

$$L_i^{\mathcal{U}} = f\Bigg(\sum_{j\in\mathcal{N}_i^+} w_{ij}^+ LF_j^{\mathcal{V}} + \sum_{j\in\mathcal{N}_i^-} w_{ij}^- LF_j^{\mathcal{V}}\Bigg)$$

$$LF_j^{\mathcal{V}} = f\Bigg(\sum_{i\in\mathcal{N}_j^+} w_{ij}^+ L_i^{\mathcal{U}} + \sum_{i\in\mathcal{N}_j^-} w_{ij}^- L_i^{\mathcal{U}}\Bigg)$$

where $f(.)$ denotes the normalization function $f(x) = \frac{2}{1+exp(x)} - 1$ such that $L_i^{\mathcal{U}}, L_j^{\mathcal{V}} \in [-1,1]$, and $w^{\pm}$ and $\mathcal{N}^{\pm}$ are defined as before.

We use the equations above to iteratively update the liberality values of the network objects. Those nodes with unknown labels are initially assigned values $L_i^{\mathcal{U}} = L_j^{\mathcal{V}} = \epsilon \approx 0$, i.e. unbiased priors. For convergence, we set a maximum number of iterations or use an ϵ-stopping criterion.

The sign of the liberality scores are used to classify; e.g. for people $\mathcal{L}_{u_i} \leftarrow L$ if $L_i^{\mathcal{U}} > 0$ and $\mathcal{L}_{u_i} \leftarrow C$ otherwise. To rank we sort by the liberality scores, e.g. $r^{\mathcal{U}}_{HITS} \leftarrow sort(L_i^{\mathcal{U}})$ orders people from most conservative to most liberal.

Signed Spectral Clustering

Spectral clustering (Ng, Jordan, and Weiss 2001) uses the second smallest positive eigenvector of the Laplacian matrix of a given graph to partition it into two clusters. The Laplacian matrix is built using the adjacency matrix of the graph, which often consists of all non-negative entries. One way to modify it for signed graphs is to build an augmented adjacency matrix, where the first part represents the positive entries, and the second part represents the negative entries.

In particular, we build $A = [A^+ A^-] \in \mathbb{R}^{n\times 2m}$ where

$$a_{ij}^+ = \begin{cases} a_{ij}, & \text{if } a_{ij} > 0\,, \\ 0, & \text{otherwise} \end{cases} \quad a_{ij}^- = \begin{cases} -a_{ij}, & \text{if } a_{ij} < 0\,, \\ 0, & \text{otherwise} \end{cases}$$

such that all the entries of A are non-negative.

Next we construct $\tilde{A} = \begin{pmatrix} 0_{n\times n} & A \\ A^T & 0_{2m\times 2m} \end{pmatrix}$ and define the Laplacian of $\tilde{A}$ as $L = D - \tilde{A}$ where D is the diagonal matrix with entries $d_{ii} = \sum_{j=1}^{n+2m} \tilde{a}_{ij}$. We use the normalized Laplacian $\bar{L} = D^{-1/2}LD^{-1/2}$, and compute its second eigenvector $v_2 \in \mathbb{R}^{(n+2m)\times 1}$ that corresponds to the second smallest eigenvalue.

For partitioning, we use the first n entries $v_2^{1:n}$ for people and the next m entries $v_2^{n+1:n+m}$ for subjects, where negative and positive values determine the two clusters c_1 and c_2 for each set of objects; e.g. for people $u_i \in c_1$ if $v_{2,i}^{1:n} > 0$ and $u_i \in c_2$ otherwise. We assign the two class labels to the two clusters to achieve maximum agreement with ground truth.[3] We also use these entries for ranking; e.g. for people $r^{\mathcal{U}}_{Spec} \leftarrow sort(v_2^{1:n})$.

Experiments

To evaluate the effectiveness of the proposed method in comparison to the competitor methods, we used both synthetic and real datasets. As measures of effectiveness we consider both (1) the accuracy in polarity *prediction*, and (2) the accuracy in polarity *ranking*.

Synthetic data

To study the behavior of the methods, we created several synthetic bipartite graphs $G(U, V, E)$, $|U| = n$, $|V| = m$, with nodes having various degree and polarity distributions.

First, we sampled degree values $1 \leq d(u_i) \leq m$ and $1 \leq d(v_j) \leq n$ from probability distribution P_d. We truncated the values to lie in correct ranges and rounded to nearest integers. Similarly, we sampled polarity values $-1 \leq p(u_i), p(v_j) \leq +1$ from probability distribution P_p. To generate edges, we created a list of randomly permuted (u_i, v_j) pairs, where each u_i and v_j appeared in the list $d(u_i)$ and $d(v_j)$ times, respectively. We only used the largest connected component and discarded the rest. Finally, we assigned the sign of each edge $e(u_i, v_j)$ to $sign(p(u_i))sign(p(v_j))$ with probability $1 - (1-|p(u_i)|)(1-|p(v_j)|)$, and to $-sign(p(u_i))sign(p(v_j))$ with probability $(1-|p(u_i)|)(1-|p(v_j)|)$.

We experimented with the cross product of two degree distributions P_d; (1) generalized Pareto, and (2) Normal, and three polarity distributions P_p; (a) Beta, (b) Bimodal, and (c) Uniform, with varying parameters.

Accuracy measures For polarity prediction, we use the accuracy of the methods in assigning the nodes their correct polarity sign. In particular, prediction accuracy is equal to $Acc(h) = \sum_{n_i\in U\cup V} \frac{\mathbb{1}\{sign(h(n_i))=sign(p(n_i))\}}{(n+m)}$, where $h(.)$ denotes the score assigned by a given method to a given node, and $\mathbb{1}\{.\}$ is the indicator function.

In this work, not only do we care about the classification accuracy of the methods, but we also care about their effectiveness in ranking the nodes by their polarity in the spectrum.[4] For ranking accuracy, we need a measure that accounts for the position of the nodes in their ranked order produced by the method, which penalizes for high polarity nodes being ranked lower in the order. Discounted cumulative gain (DCG) is a widely used measure especially in information retrieval that serves this purpose. For the ranking $r(h)$ of nodes by a given method, DCG is defined as $DCG_{r(h)} = \sum_{i=1}^{m+n} \frac{2^{p(n_i)}-1}{log_2(i+1)}$. The maximum DCG value is achieved when the nodes are sorted in order by their orig-

[3] Since the two clusterings are the same up to a permutation.

[4] We use the ranking accuracy rather than absolute polarity magnitude estimation accuracy, as the scores computed by different methods are not directly comparable to polarity scores. For example, SPP and *HITS* compute probabilities.

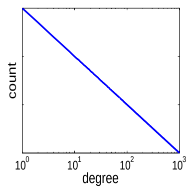
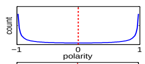
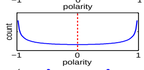
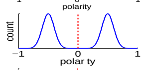
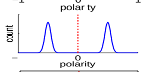
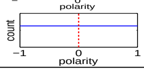
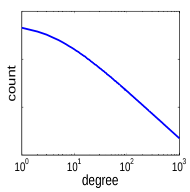
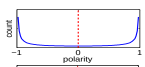
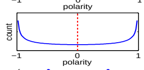
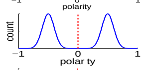
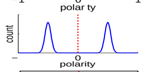
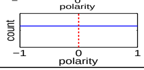
Degree distribution		Polarity distribution	Accuracy of methods SPP	*Spec*	*HITS*	*wvRN*	nDCG of methods SPP	*Spec*	*HITS*	*wvRN*	baselines *min*	*rand*
Pareto(1,1,1) 		$\beta(a,b:0.1)$	**0.945**	0.509	**0.560**	0.440	**0.985**	0.879	0.889	**0.948**	0.764	0.880
		$\beta(a,b:0.5)$	**0.829**	0.517	0.532	**0.868**	**0.965**	0.882	0.893	**0.949**	0.777	0.883
		$bi\mathcal{N}(\sigma:0.1)$	**0.720**	0.508	**0.755**	0.600	**0.971**	0.919	**0.970**	0.936	0.837	0.916
		$bi\mathcal{N}(\sigma:0.05)$	**0.703**	0.515	**0.557**	0.501	**0.976**	0.925	**0.927**	0.925	0.848	0.923
		*Uniform*	**0.704**	0.505	**0.747**	0.646	**0.958**	0.891	**0.955**	0.918	0.798	0.893
		avg	**0.780**	0.510	0.630	0.611	**0.971**	0.899	0.927	0.935	0.805	0.899
		±std	0.105	0.005	0.110	0.164	0.010	0.021	0.036	0.014	0.037	0.019
Pareto(1,5,1) 		$\beta(a,b:0.1)$	**0.992**	**0.532**	0.507	0.494	**0.987**	0.890	0.869	**0.955**	0.769	0.882
		$\beta(a,b:0.5)$	**0.905**	0.897	0.505	**0.950**	**0.973**	**0.971**	0.873	0.970	0.781	0.885
		$bi\mathcal{N}(\sigma:0.1)$	0.829	0.558	**0.884**	**0.867**	**0.980**	0.951	**0.981**	0.965	0.836	0.915
		$bi\mathcal{N}(\sigma:0.05)$	**0.850**	0.511	0.503	**0.856**	**0.989**	0.932	0.917	**0.973**	0.851	0.924
		*Uniform*	**0.813**	0.505	**0.863**	0.530	**0.965**	0.888	**0.966**	0.934	0.798	0.893
		avg	**0.878**	0.601	0.652	0.739	**0.979**	0.926	0.921	0.959	0.807	0.900
		±std	0.073	0.167	0.202	0.211	0.010	0.037	0.051	0.016	0.035	0.019

Figure 1: Classification (left) and ranking (right) accuracies of four competing methods at changing polarities, when degree distribution follows (top) generalized $Pareto$ (1,1,1), and (bottom) generalized $Pareto$ (1,5,1).

inal polarity values $p(n_i)$, denoted as iDCG. We normalize the above with iDCG such that $nDCG = \frac{DCG}{iDCG} \in [0, 1]$.

Since our goal is to rank highly positive polarity nodes at the top and highly negative polarity nodes at the bottom, positions of both ends of the ordered list matter. To account for that, we compute two nDCG scores for each method; one where the nodes are ranked in *desc*ending order of their $h(.)$ values with original polarity values being $\frac{1+p}{2} \in [0, 1]$, and another with an *asc*ending ordering and polarity values flipped, i.e. $\frac{1-p}{2} \in [0, 1]$. The final ranking accuracy is $nDCG_{r(h)} = (nDCG^{desc}_{r(h)} + nDCG^{asc}_{r(h)})/2 \in [0, 1]$.

Results The skewness of the polarity scores determines the difficulty of the prediction and the ranking tasks. Intuitively, the tasks become easier if many nodes have polarity scores close to -1 or $+1$. We generated polarity scores of varying skewness using three distributions P_p as shown in Figures 1&2. Beta distribution $\beta(a, b)$ generates highly skewed values for smaller a, b. Bimodal $bi\mathcal{N}(\sigma)$ generates two normal distributions centered at -0.5 and 0.5, respectively with varying standard deviation σ. Finally, $uniform$ distribution produces the least skewness in scores.

We generated graphs with two different degree distributions; generalized $Pareto(K, \sigma, \theta)$ where K denotes the weight of the tail, σ the scale, and θ the minimum threshold, as well as $\mathcal{N}ormal(\mu, \sigma)$. Pareto creates highly skewed power-law-like degree distributions as observed in many real-world graphs. In our experiments we set the number of nodes $n = m = 1000$. For Pareto we used $K = 1$, $\theta = 1$, and $\sigma = \{1, 5\}$, and for Normal $\mu = 100$ and $\sigma = \{20, 40\}$. Various σ parameters are used to evaluate the effect of the degree variance on the performance of the algorithms.

For each degree and polarity distribution, we compared the performance (in classification and ranking) of our SPP method and the three competitor algorithms *Spec* (Ng, Jordan, and Weiss 2001), *HITS* (Kleinberg 1998), and *wvRN* (Macskassy and Provost 2003), as described in §.

Figure 1 (top) shows the classification accuracy and the nDCG ranking accuracy results for degree distribution $Pareto(1, 1, 1)$ and varying polarities. The last two columns of the figure depict nDCG scores of two baselines; first (min) is the lower bound when the reverse of the optimal ordering is used, and the second ($rand$) is the average nDCG of 100 random orderings.[5]

We observe that SPP is one of the top two performing methods (shown in bold-face) in all cases and its accuracy is always above 0.7. While performing well at certain polarities, the accuracies of *HITS* and *wvRN* fluctuate significantly. On the other hand, the accuracy of SPP drops gradually with decreasing skewness in polarities as one might expect. An important result to note here is that *Spec* performs very poorly at all polarity levels—this could be due to its search for a good "cut" in the graph which often does not exist for graphs with highly skewed degree distributions.

Similar conclusions are drawn from Figure 1 (bottom) for degree distribution $Pareto(1, 5, 1)$. Increasing the variance

[5]Standard deviations ranged between 1×10^{-4} and 5×10^{-4}.

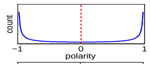
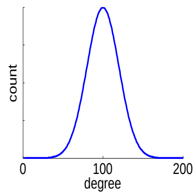
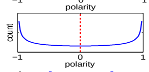
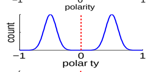
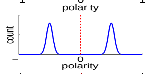
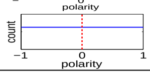
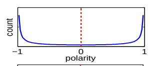
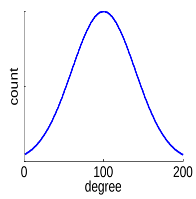
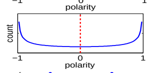
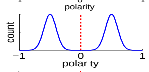
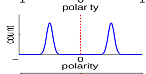
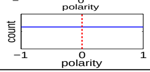
Degree distribution	Polarity distribution		Accuracy of methods SPP	Spec	HITS	wvRN	nDCG of methods SPP	Spec	HITS	wvRN	baselines min	rand
$\mathcal{N}(100, 20)$		$\beta(a,b:0.1)$	**1.000**	**1.000**	0.500	**1.000**	0.987	**0.993**	0.942	**0.995**	0.769	0.883
		$\beta(a,b:0.5)$	**1.000**	**1.000**	0.500	0.500	**0.963**	**0.991**	0.839	0.961	0.779	0.884
		$bi\mathcal{N}(\sigma:0.1)$	0.996	**0.999**	0.500	**0.998**	0.982	**0.992**	0.883	**0.991**	0.838	0.917
		$bi\mathcal{N}(\sigma:0.05)$	**1.000**	**1.000**	0.500	0.556	**0.991**	**0.994**	0.959	0.964	0.851	0.924
		*Uniform*	**0.956**	**0.994**	0.482	0.504	0.961	**0.991**	0.860	**0.963**	0.800	0.894
		avg	0.990	**0.997**	0.496	0.712	0.977	**0.992**	0.897	0.975	0.807	0.900
		±std	0.019	0.003	0.008	0.263	0.014	0.001	0.052	0.017	0.036	0.019
$\mathcal{N}(100, 40)$		$\beta(a,b:0.1)$	**1.000**	**1.000**	**1.000**	**1.000**	0.985	**0.991**	0.985	**0.994**	0.769	0.883
		$\beta(a,b:0.5)$	**0.998**	**0.999**	0.500	0.499	**0.964**	**0.989**	0.932	0.956	0.781	0.885
		$bi\mathcal{N}(\sigma:0.1)$	0.994	0.996	**0.997**	**0.998**	0.983	**0.991**	0.986	**0.988**	0.838	0.917
		$bi\mathcal{N}(\sigma:0.05)$	0.997	0.996	**0.999**	**0.999**	0.992	**0.994**	0.992	**0.994**	0.850	0.924
		*Uniform*	0.968	**0.996**	0.983	**0.999**	0.962	**0.990**	0.977	**0.981**	0.796	0.892
		avg	0.991	**0.997**	0.896	0.899	0.977	**0.991**	0.974	0.983	0.807	0.900
		±std	0.013	0.002	0.221	0.224	0.013	0.002	0.024	0.016	0.035	0.019

Figure 2: Classification accuracy (left) and ranking accuracy (right) of four competing methods at changing polarities, when degree distribution follows (top) $Normal$ (100, 20), and (bottom) $Normal$ (100, 40).

($\sigma = 5$) yields a less skewed degree distribution. In effect, the classification accuracies of all methods increase—classifying high degree nodes is easier as more neighbors provide more information.

With respect to the ranking accuracies (nDCG) of the methods, we observe that SPP achieves significantly better ranking than random ordering at all polarity levels, which does not hold for the competitor methods. The nDCG values of *Spec* and *HITS* are close to that of random ordering when their prediction accuracies are also close to random. Interestingly, we notice that the nDCG of *wvRN* can be quite high while its prediction accuracy is quite low (e.g. first row in Fig. 1 (top)). Further analysis showed that the converged scores of *wvRN* are highly concentrated around 0, and while the relative ordering of nodes could be mostly correct, their signs are often mispredicted.

In Figure 2 we use $Normal$ degree distribution ($\mu = 100$) with $\sigma = 20$ (top) and $\sigma = 40$ (bottom). With far fewer low degree nodes, the accuracy of SPP increases significantly at all polarities. As before, the performances of *HITS* and *wvRN* largely oscillate, and the accuracies of all methods increase when the degree variance is increased, as we observed with $Pareto$ (more neighbors, more information). SPP continues to perform well with very high accuracies. Note that *Spec* starts performing well on these graphs with more uniform degrees; unlike scale-free graphs these have better "cut"s. However, *Spec* can be slow for graphs with high density, since it relies on eigenvector computation. We compare the scalability of *Spec* to SPP in the next section.

Overall, SPP performs consistently well at a variety of degree distributions and polarity levels, and proves to be the most reliable method among all the competing methods.

Real-world data

We also validated our method's performance on two real-world datasets, *US Congress*[6] and *PolForum*[7]. Both of these datasets are crawled from public data and are available at the indicated URLs.

US Congress consists of the roll call votes for the $111th$ US Congress. It includes The Senate and The House of Representatives in years 2009-10. The $111th$ Senate has 108 senators, and The $111th$ House has 451 representatives. The data contains the senators' votes on 696 bills and the representatives' votes on 1655 bills, respectively. The signed bipartite graphs we built from this data consist of congressmen versus the bills, in which a signed edge denotes the 'yes' $(+1)$ or 'no' (-1) roll-call vote of a congressman for a bill, where the absence of an edge (0) denotes non-attendance.

PolForum consists of a crawl from a political forum where users write posts on their opinion about various political issues such as religion, gay&lesbian rights, gun control, etc. The edges in the signed bipartite graph of users versus political issues denote the 'in favor' $(+1)$ or 'against' (-1) opin-

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

[7]http://www.politicalforum.com/forum.php

ion of a user on an issue, where no edge (0) denotes no posts of the user on the particular issue.

To infer the signs of edges, i.e. whether a user is 'in favor'/'for' or 'against' an issue, we first performed sentiment analysis on the posts. The results of sentiment scoring, however, were unsatisfactory.[8] This may be due to the complexity of natural language (e.g. ambiguity, sarcasm) encountered in such discussion forums. Thus we resorted to manual labeling.[9] Table 2 provides a summary of our datasets.

Table 2: Real datasets used in this work. D: democrat, R: republican, L: liberal, C: conservative.

Datasets	U Nodes	V Nodes	Edges
US Senate	64 (D) + 42 (R) Senators	696 Bills	66416 yes/no Votes
US House	268 (D) + 183 (R) Representatives	1655 Bills	680930 yes/no Votes
PolForum	7 (L) + 5 (C) Users	21 Issues	72 for/against Opinions

Results Both of the real datasets contain ground truth polarity labels; Conservative (-1) and Liberal $(+1)$ for the political forum users, and Republican (-1) and Democrat $(+1)$ for the congressmen.

Classification Accuracy: Table 3 shows the prediction accuracy of all the methods. We notice that sPP performs quite well on these datasets with near perfect accuracy. *Spec* also achieves high accuracies across all datasets. The node degrees of our real graphs are quite uniform—for example almost all congressmen are expected to vote on a bill. As verified by the synthetic data experiments in the previous section, *Spec* performs well on such graphs. In comparison, sPP achieves the highest accuracy across all datasets.

Table 3: Classification accuracies of the methods.

Methods	*US Senate*	*US House*	*PolForum*
sPP	**1.0000**	**0.9933**	**1.0000**
Spec	**1.0000**	0.9911	**1.0000**
HITS	0.9717	0.6497	0.9167
wvRN	**1.0000**	0.5965	0.9167

Running time: While *Spec* achieves comparable accuracy to sPP on the real datasets, the running time of *Spec* grows significantly larger than that of sPP with increasing graph size. As shown in Figure 3, sPP scales only *linearly* with respect to the number of edges in the graph, which means that it is suitable for very large graphs.

Sensitivity: Next we study the sensitivity of the classification accuracy to the ϵ parameter used in our method (see Table 1). In Figure 4 we depict the sPP accuracy for various $\epsilon \in [0, 0.5]$. The results show that the accuracies remain consistently high for all of the datasets.

Ranking Accuracy: Next we evaluate our method in predicting the *extent* of polarization in the real datasets, i.e. its ranking performance. Many political scientists rely on DW-NOMINATE scores (Poole and Rosenthal 1991) to assign a single liberal-conservative score to each congressperson. At the heart of the NOMINATE methods are algorithms that analyze preferential and choice data, such as legislative roll-call voting behavior, and utilize those to arrange individuals and choices in a two-dimensional space. Most of the variation in voting patterns are explained by placement along the liberal-conservative first dimension. Therefore, we downloaded the DW-NOMINATE scores for the $111th$ Senate and the House[10], and considered the first dimension of these DW-NOMINATE scores as the ground truth polarity scores for the congressmen. We use these scores to evaluate the ranking of our method.

In Figure 5 we demonstrate that sPP correlates with this most-cited roll-call method DW-NOMINATE at extremely high levels. The scatter plots depict the ranks produced by DW-NOMINATE on the horizontal axes and sPP on the vertical axes[11]—the congressmen tend to cluster on the 45 degree diagonal line, indicating that the two measures are highly similar. Specifically, Spearman's rank correlation between sPP and DW-NOMINATE is 0.94 for *US Senate*, and 0.89 for *US House*. Compared to DW-NOMINATE, however, sPP is easier to implement and compute. As such, it is noteworthy that sPP has remarkably high level of correlation with DW-NOMINATE, despite lacking the complex machinery of the prevailing method. Moreover, sPP is general and is not limited in application to only congressional data, whereas DW-NOMINATE scores are specifically crafted for such data.

While we were able to attain ground truth scores on polarity for congressmen, there does not exist such scores neither

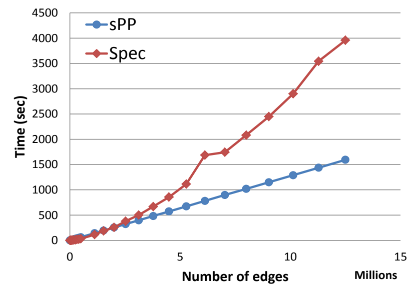


Figure 3: Run time of sPP and *Spec* versus number of edges. sPP scales linearly w.r.t. graph size.

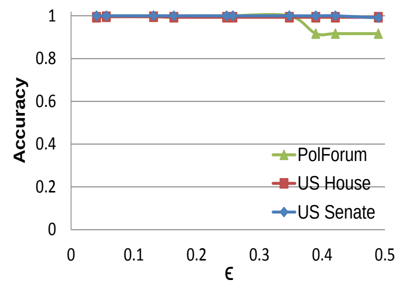


Figure 4: Classification accuracy of sPP for changing ϵ parameter remains stable across all datasets.

[8]We used http://www.clips.ua.ac.be/pages/pattern-en, which assigns a senti-score $[-1, 1]$ to a given text and assigned the sign of the sum of senti-scores of all posts of a user for an issue. Manual analysis, however, revealed many incorrect assignments.

[9]We labeled edges as in favor or against, and users as conservative or liberal.

[10]http://ibm.co/1aQOmzr, http://ibm.co/110BLDd

[11]A rank of 1 denotes the most liberal congressman.

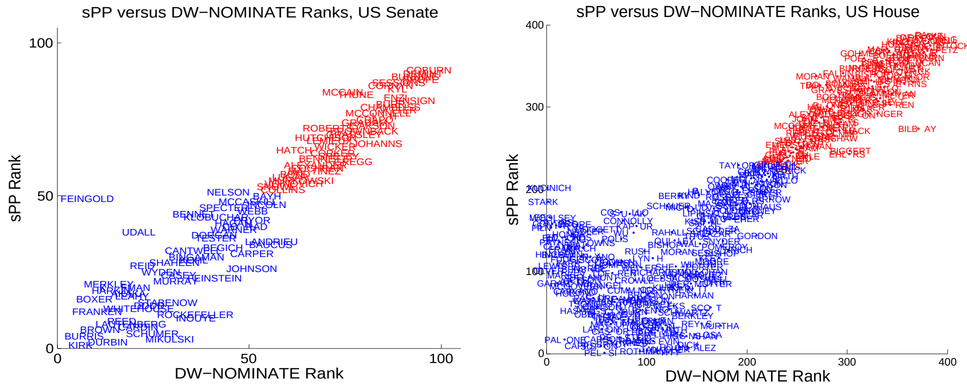
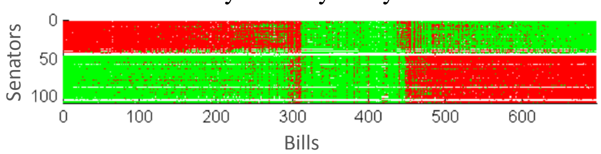


Figure 5: Congressmen ranks indicated by DW-NOMINATE (x axes) and our sPP (y axes) for the $111th$ (left) *US Senate* and (right) *US House*. The two methods produce comparable ranks: Spearman's rank correlation is respectively 0.94 and 0.89. Separation of parties is also clear from the rankings. Democrats: blue, Republicans: red (figures best in color).

for the corresponding bills nor the political issues or forum users in our *PolForum* dataset. To study the ranking performance of our method on those, we exploit visualization.

We depict the adjacency matrix of the *US Senate* bipartite graph below, where the rows (senators) and the columns (bills) are ordered in increasing order of their democratic partisanship. The figure clearly reveals the two political groups of congressmen in the Senate, the two main groups of bills favored by these groups, as well as the "unifying" bills which were mostly voted 'yes' by all the senators.[12]



Finally, we show the adjacency matrix of *PolForum* rank-ordered by sPP in Figure 6. For this dataset, the ordering of the political issues reveals interesting insights. We observe that religion and abortion lie in opposite ends of the spectrum, mostly favored by conservative and liberal users, respectively. Liberal users also strongly advocate for gay&lesbian rights. On the other hand, gun control is found to reside in the gray zone—there is not much consensus among liberal users on the issue of gun control. Analysis of forum posts by these users showed that while several argued for it, others argued against it by opposing the restriction of civil liberties. This lack of agreement on gun control is not a freak occurrence. For example, roll-call votes on gun control also routinely split party coalitions in the Congress, with socially conservative Democrats joining most Republicans in opposing more regulation and socially liberal Republicans joining most Democrats in supporting gun control.[13]

Conclusion

We have addressed the problem of political polarity prediction and ranking. Our solution revolves around using a *signed* bipartite network representation of individuals asserting +/− opinions on political subjects, and formulating the problem as a node classification task on opinion networks. We proposed a linear algorithm that exploits network effects to learn both the polarity labels as well as the rankings of people and issues in a completely unsupervised manner. We compared our method[14] to three well-known algorithms, weighted-vote relational classifier (Macskassy and Provost 2003), HITS (Kleinberg 1998), and spectral clustering (Ng, Jordan, and Weiss 2001), adapted to handle signed edges. Experiments on synthetic and real data showed the effectiveness and scalability of our approach on a variety of opinion networks, for both classification and ranking. Future work can look at changes in polarization over time, performing incremental updates of the beliefs on dynamic networks.

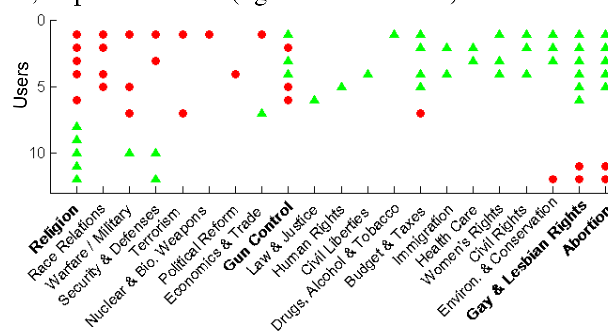


Figure 6: Adjacency matrix of *PolForum* after rows (Users) and columns (Issues) are ordered in sPP ranking order. The green triangles/red circles denote 'for'/'against' opinions.

Acknowledgements

We thank the reviewers for helping us improve our manuscript. This material is based on work supported by the Army Research Office under Contract No. W911NF-14-1-0029, Stony Brook University Office of Vice President for Research, and a gift from Northrop Grumman Aerospace Systems. Any findings and conclusions expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding parties.

[12]Similar results for *US House* are omitted for brevity.

[13]http://en.wikipedia.org/wiki/NOMINATE_(scaling_method)

[14]Our real-world datasets and ground-truth labels/polarity scores are available at http://www.cs.stonybrook.edu/~leman/pubs.html#data.

References

Adamic, L. A., and Glance, N. 2005. The political blogosphere and the 2004 U.S. election: divided they blog. In *LinkKDD*, 36–43.

Adar, E.; Zhang, L.; Adamic, L. A.; and Lukose, R. 2004. Implicit structure and the dynamics of blogspace. In *Workshop on the Weblogging Ecosystem*.

Awadallah, R.; Ramanath, M.; and Weikum, G. 2010. Language-model-based pro/con classification of political text. In *SIGIR*, 747–748.

Balasubramanyan, R.; Cohen, W. W.; Pierce, D.; and Redlawsk, D. P. 2012. Modeling polarizing topics: When do different political communities respond differently to the same news? In *ICWSM*.

Brin, S., and Page, L. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* 30(1–7):107–117.

Chiang, K.-Y.; Natarajan, N.; Tewari, A.; and Dhillon, I. S. 2011. Exploiting longer cycles for link prediction in signed networks. In *CIKM*, 1157–1162.

Cohen, R., and Ruths, D. 2013. Classifying political orientation on twitter: It's not easy! In *ICWSM*.

Conover, M.; Gonçalves, B.; Ratkiewicz, J.; Flammini, A.; and Menczer, F. 2011a. Predicting the political alignment of twitter users. In *IEEE Conference on Social Computing (SocialCom)*.

Conover, M.; Ratkiewicz, J.; Francisco, M.; Gonçalves, B.; Flammini, A.; and Menczer, F. 2011b. Political polarization on twitter. In *ICWSM*.

DuBois, T.; Golbeck, J.; and Srinivasan, A. 2011. Predicting trust and distrust in social networks. In *SocialCom/PASSAT*, 418–424.

Getoor, L.; Friedman, N.; Koller, D.; Pfeffer, A.; and Taskar, B. 2007. Probabilistic relational models. In *Introduction to Statistical Relational Learning*. MIT Press.

Godbole, N.; Srinivasaiah, M.; and Skiena, S. 2007. Large-scale sentiment analysis for news and blogs. In *ICWSM*.

Golbeck, J., and Hansen, D. L. 2011. Computing political preference among twitter followers. In *CHI*, 1105–1108.

Guerra, P. H. C.; Jr., W. M.; Cardie, C.; and Kleinberg, R. 2013. A measure of polarization on social media networks based on community boundaries. In *ICWSM*.

Guha, R.; Kumar, R.; Raghavan, P.; and Tomkins, A. 2004. Propagation of trust and distrust. In *WWW*, 403–412.

He, Y.; Saif, H.; Wei, Z.; and Wong, K. 2012. Quantising opinons for political tweets analysis. In *LREC*.

Huang, J.; Nie, F.; Huang, H.; and Tu, Y.-C. 2012. Trust prediction via aggregating heterogeneous social networks. In *CIKM*.

Kindermann, R., and Snell, J. L. 1980. *Markov Random Fields and Their Applications*.

Kleinberg, J. M. 1998. Authoritative sources in a hyperlinked environment. In *ACM-SIAM Symposium on Discrete Algorithms*.

Leskovec, J.; Huttenlocher, D. P.; and Kleinberg, J. M. 2010a. Predicting positive and negative links in online social networks. In *WWW*, 641–650. ACM.

Leskovec, J.; Huttenlocher, D. P.; and Kleinberg, J. M. 2010b. Signed networks in social media. In *CHI*, 1361–1370. ACM.

Lin, F., and Cohen, W. W. 2008. The multirank bootstrap algorithm: Self-supervised political blog classification and ranking using semi-supervised link classification. In *ICWSM*.

Livne, A.; Simmons, M. P.; Adar, E.; and Adamic, L. A. 2011. The party is over here: Structure and content in the 2010 election. In *ICWSM*.

Lo, D.; Surian, D.; Prasetyo, P. K.; Zhang, K.; and Lim, E.-P. 2013. Mining direct antagonistic communities in signed social networks. *Inf. Process. Man.* 49(4):773–791.

Macskassy, S., and Provost, F. 2003. A simple relational classifier. In *2nd Workshop on Multi-Relational Data Mining, KDD*, 64–76.

McGlohon, M.; Glance, N. S.; and Reiter, Z. 2010. Star quality: Aggregating reviews to rank products and merchants. In *ICWSM*.

Neville, J., and Jensen, D. 2003. Collective classification with relational dependency networks. In *KDD*.

Newman, M. E. J. 2006. Modularity and community structure in networks. *PNAS* 103(23):8577–82.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. In *NIPS*, 849–856.

Pak, A., and Paroubek, P. 2010. Twitter as a corpus for sentiment analysis and opinion mining. *Proceedings of LREC*.

Pandit, S.; Chau, D. H.; Wang, S.; and Faloutsos, C. 2007. Netprobe: a fast and scalable system for fraud detection in online auction networks. In *WWW*, 201–210.

Pennacchiotti, M., and Popescu, A.-M. 2011. A machine learning approach to twitter user classification. In *ICWSM*.

Poole, K. T., and Rosenthal, H. 1991. Patterns of Congressional Voting. *AJPS* 35(1):228–278.

Symeonidis, P.; Tiakas, E.; and Manolopoulos, Y. 2010. Transitive node similarity for link prediction in social networks with positive and negative links. In *RecSys*, 183–190. ACM.

Thomas, M.; Pang, B.; and Lee, L. 2006. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *EMNLP*.

Traag, V. A., and Bruggeman, J. 2009. Community detection in networks with positive and negative links. *Phys. Rev. E* 80:036115.

Tumasjan, A.; Sprenger, T. O.; Sandner, P. G.; and Welpe, I. M. 2010. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *ICWSM*. The AAAI Press.

Watanabe, Y. 2011. Uniqueness of belief propagation on signed graphs. In *NIPS*, 1521–1529.

Yang, S.-H.; Smola, A. J.; Long, B.; Zha, H.; and Chang, Y. 2012. Friend or frenemy?: predicting signed ties in social networks. In *SIGIR*, 555–564.

Yedidia, J. S.; Freeman, W. T.; and Weiss, Y. 2003. Understanding belief propagation and its generalizations. In Exploring AI in the new millennium. 239–269.

Zhang, K.; Lo, D.; Lim, E.-P.; and Prasetyo, P. K. 2013. Mining indirect antagonistic communities from social interactions. *Knowl. Inf. Syst.* 35(3):553–583.

Zhou, D. X.; Resnick, P.; and Mei, Q. 2011. Classifying the political leaning of news articles and users from user votes. In *ICWSM*.