

# On Ranking Senators By Their Votes

Mugizi Rwebangira  
Howard University  
Systems and Computer Science  
2300 6th Street Washington, DC 20059  
rweba@cs.cmu.edu

August 6, 2009

## Abstract

The problem of ranking a set of objects given some measure of similarity is one of the most basic in machine learning. Recently [1] proposed a method based on techniques in semi-supervised learning utilizing the graph Laplacian. In this work we consider a novel application of this technique to ranking binary choice data and apply it specifically to ranking US Senators by their ideology.

## 1 Introduction

Ranking is one of the most fundamental problems in machine learning. Over the years, many algorithms have been proposed many of them relying on classical techniques such as linear regression. In order to apply regression techniques typically we need have access to the features of the objects in order to learn the regression equation. We can then run the regression equation on the test data and output our ranking.

Recently there has been more interest in the setting where we only have access to the similarity among objects and we want to output a ranking. The natural interpretation of this ranking is that we want similar objects to be ranked close to each other and dissimilar objects to be ranked far apart from each other. This set of similarities can be visualized as a graph, where the nodes are the objects we want to rank and the weight on the edges expresses the similarity of the examples connected by that edge.

The “graph learning” setting has proven very popular in machine learning and there have been a slew of papers exploring this paradigm [1, 2, 6, 7]. In a recent

paper Agarwal [1] proposed applying the graph learning technique to the problem of ranking. In this paper we propose to apply this technique to ranking binary choice data, with the specific application of ranking US Senators by their ideology.

## 2 Background

### 2.1 Ranking Legislators

The problem of ranking politicians by ideology has been well studied in the political science literature. Currently the most popular such ranking system is VoteView developed by political scientist Keith Poole and hosted on his web site VoteView.com. Voteview works by projecting the set of legislators into a low dimensional Euclidean space and then iteratively searching for cutting planes that optimally divide the legislators into sets that agree with each other [4].

In particular we note that this optimization procedure is not guaranteed to converge to a global minimum. By contrast the method that we present involves only basic linear algebra and is guaranteed to give a global minimum. In addition it is very easy to implement and has a lower running time.

### 2.2 The Combinatorial Graph Laplacian

In our solution we will use a matrix known as the *combinatorial graph Laplacian* (or simply the *Laplacian* of a graph) because of several suggestive analogies with the classical Laplacian differential operator widely used in physics (among other places). The Laplacian has many interesting properties of which we will exploit only a few in this work. For more details refer to the thesis of Zhu[6].

For an undirected graph  $G$ , the Laplacian  $\Delta$  is defined as the matrix

$$\Delta = D - A$$

where  $A$  is the adjacency matrix of  $G$  and  $D$  is the diagonal matrix which has the degree of each vertex on the diagonal (i.e the degree matrix which is equivalently the sum of very row).

The concept can be straightforwardly generalized to a weighted graph by defining

$$\Delta = D - W$$

where  $W_{i,j}$  is the weight between nodes  $i$  and  $j$  and  $D_{i,i}$  is the sum of row  $i$  of matrix  $W$ .

### 3 Ranking

Our goal is to develop a method for ranking legislators by ideology based on their votes. We can use a simple model for this situation: **Each senator has a ideology score  $f$  between  $-1$  and  $+1$  that represents their political leanings.** For example we can let  $+1$  represent an extremely liberal senator and  $-1$  an extremely conservative one. Our task is to compute these scores based on their votes.

To derive an algorithm we make a simple and obvious assumption: **Senators that have a similar voting record should have similar ideological scores.** That is if two senators voting records are very similar we assume their political views are also very similar.

We propose the following as our objective function which we will seek to minimize.

$$\mathcal{E}(f) = \sum_i^n \sum_j^n W_{i,j} (f_i - f_j)^2$$

where  $W_{i,j}$  is the similarity between senators  $i$  and  $j$  and  $n$  is the number of senators.

To obtain a ranking we have to find the values of  $f_i$  which minimize this objective function.

#### Lemma 1

$$\mathcal{E}(f) = \sum_i^n \sum_j^n W_{i,j} (f_i - f_j)^2 = 2f^T \Delta f$$

where  $\Delta$  is the graph Laplacian and  $f$  is the vector of all the  $f_i$ 's.

#### Proof:

$$\begin{aligned} \mathcal{E}(f) &= \sum_i^n \sum_j^n W_{i,j} (f_i - f_j)^2 \\ &= \sum_i^n \sum_j^n W_{i,j} (f_i^2 - 2f_i f_j + f_j^2) \\ &= \sum_i^n \sum_j^n W_{i,j} f_i^2 - 2W_{i,j} f_i f_j + W_{i,j} f_j^2 \\ &= \sum_i^n \sum_j^n W_{i,j} f_i^2 - 2 \sum_i^n \sum_j^n W_{i,j} f_i f_j + \sum_i^n \sum_j^n W_{i,j} f_j^2 \end{aligned}$$

Now

$$\sum_i^n \sum_j^n W_{i,j} f_i^2 = \sum_i^n (W_{i,1} + W_{i,2} + \dots) f_i^2 = f^T D f$$

Where  $D$  is a diagonal matrix and  $D_{i,i}$  is the sum of row  $i$  of matrix  $W$ .

Similarly

$$\sum_i^n \sum_j^n W_{i,j} f_j^2 = f^T D f$$

Here we use the assumption that matrix  $W$  is symmetric.

Lastly

$$\sum_i^n \sum_j^n W_{i,j} f_i f_j = f^T W f$$

Putting it all together

$$\begin{aligned} &= \sum_i^n \sum_j^n W_{i,j} f_i^2 - 2 \sum_i^n \sum_j^n W_{i,j} f_i f_j + \sum_i^n \sum_j^n W_{i,j} f_j^2 \\ &= f^T D f - 2f^T W f + f^T D f = 2f^T (D - W) f = 2f^T \Delta f \end{aligned}$$

#### QED

In order to get non-trivial results we will have to specify the values of at least two of the  $f_i$ . (Otherwise the minimum can be obtained by just setting all the  $f_i$  to zero which is not very useful). It is convenient to set one  $f_i$  to  $+1$  and another  $f_i$  to  $-1$ , corresponding to the most ideologically pure legislators.

Once we have done this we can split up the vector  $f$  into  $f_L$  and  $f_U$  corresponding to the labeled  $f_i$  and the unlabeled  $f_i$  respectively. Likewise we can rearrange and split up  $\Delta$  into  $\Delta_{UU}$   $\Delta_{UL}$   $\Delta_{LU}$   $\Delta_{LL}$  as can be seen in the following table:

	L	U
L	$\Delta_{LL}$	$\Delta_{LU}$
U	$\Delta_{UL}$	$\Delta_{UU}$

Our task now is to minimize  $2f^T \Delta f$  which is the same as minimizing  $f^T \Delta f$

**Lemma 2** The  $f_U$  that minimizes  $f^T \Delta f$  is equal to  $-\Delta_{UU}^{-1} \Delta_{UL} f_L$

**Proof:**

$$f^T \Delta f = [f_L^T f_U^T] \begin{pmatrix} \Delta_{LL} & \Delta_{LU} \\ \Delta_{UL} & \Delta_{UU} \end{pmatrix} \begin{bmatrix} f_L \\ f_U \end{bmatrix}$$

$$= f_L^T \Delta_{LL} f_L + f_L^T \Delta_{LU} f_U + f_U^T \Delta_{UL} f_L + f_U^T \Delta_{UU} f_U$$

Differentiating by  $f_U$  (since  $f_L$  is a constant) and setting to 0 we get:

$$2\Delta_{UL} f_L + 2\Delta_{UU} f_U = 0$$

Rearranging we get

$$f_U = -\Delta_{UU}^{-1} \Delta_{UL} f_L$$

**QED**

## 4 Algorithm

To summarize, our algorithm is as follows:

1. First we compute the similarities between all pairs of examples. This will give us a similarity matrix  $W$ .
2. We then compute the graph Laplacian  $\Delta = D - W$
3. We specify at least two labeled  $f_i$  to get  $f_L$ .
4. We compute  $f_U = -\Delta_{UU}^{-1} \Delta_{UL} f_L$  to obtain the final ranking.

Now we just need to specify how we are going to compute the similarities and how we specify the labeled  $f_i$ .

### 4.1 Computing Similarities

A legislator can essentially only do 3 things on any particular vote,

1. Vote ‘‘YES’’.
2. Vote ‘‘NO’’.
3. Fail to register any vote (e.g. absent, abstaining etc).

We encode the behavior of each legislator as a vector of integers in  $\{-1, 0, +1\}$  in the obvious way and define the ‘‘distance’’  $D$  between two legislators as the Hamming difference of their respective vectors. We then define the weight  $W_{i,j}$  as  $\frac{1}{D+1}$  and thus obtain the weight matrix  $W$ .

### 4.1.1 Example

Suppose Senator Rightwinger has voted (NO, YES, ABSTAIN) on 3 bills while Senator Leftwinger has voted (YES,NO,ABSTAIN) on the same set of bills. Then we encode their votes as the vectors  $(-1,+1,0)$  and  $(+1,-1,0)$ . The Hamming distance of the two vectors is 2 and hence we will assign the similarity between the two senators the value of  $\frac{1}{2+1} = \frac{1}{3}$ .

### 4.2 Selecting the labeled $f_i$

In essence we have to specify at least two  $f_i$  to which we can confidently assign a label. For our purposes it makes the most sense to specify the most extreme examples. There are broadly two ways of doing this

1. Use domain knowledge of the political arena - e.g. look at the most extreme legislators in other rankings by advocacy groups and other parties, look at the political ideology of the legislator’s home district and other external evidence.
2. Purely internal knowledge from the dataset - For example pick the two legislators with the highest political difference (lowest similarity).

In practice we find our algorithm is robust to any reasonable choice (i.e the rankings will not change drastically based on the method).

## 5 Experimental Results

We obtained data on roll call votes for the 2007-2008 session of the US Senate from the web site of Keith T. Poole [4]. We removed the votes on which there was more than 95% agreement as those were most likely ceremonial votes.

### 5.1 Using Domain Knowledge

In this experiment we picked Senator Russell Feingold of Wisconsin and Senator Thomas Coburn of Oklahoma as our liberal and conservative exemplars. Both of these senators have a strong reputation for exemplifying the liberal and conservative wings of their respective political parties in the Senate. As per our methodology we fixed  $f_i = 1$  for Senator Feingold and  $f_i = -1$  for Senator Coburn ran our algorithm and obtained the following results:

Rank	Name	Party
1	FEINGOLD	D
2	SANDERS	D
3	LEAHY	D
4	DURBIN	D
5	HARKIN	D
6	WYDEN	D
7	BROWN	D
8	WHITEHOUSE	D
9	CARDIN	D
10	MENENDEZ	D
11	KERRY	D
12	CANTWELL	D
13	KOHL	D
14	LAUTENBERG	D
15	KLOBUCHAR	D
16	AKAKA	D
17	MURRAY	D
18	SCHUMER	D
19	REED	D
20	BOXER	D
21	BINGAMAN	D
22	LEVIN	D
23	STABENOW	D
24	REID	D
25	CASEY	D
26	MIKULSKI	D
27	FEINSTEIN	D
28	NELSON	D
29	WEBB	D
30	SALAZAR	D
31	TESTER	D
32	INOUYE	D
33	ROCKEFELLER	D
34	KENNEDY	D
35	CONRAD	D
36	DODD	D
37	DORGAN	D
38	CARPER	D
39	BAUCUS	D
40	BIDEN	D
41	MCCASKILL	D
42	LINCOLN	D
43	BYRD	D
44	CLINTON	D
45	LIEBERMAN	D
46	PRYOR	D
47	BAYH	D
48	OBAMA	D
49	LANDRIEU	D
50	NELSON	D
51	JOHNSON	D

Rank	Name	Party
52	SNOWE	R
53	COLLINS	R
54	SPECTER	R
55	SMITH	R
56	COLEMAN	R
57	WICKER	R
58	VOINOVICH	R
59	THOMAS	R
60	STEVENS	R
61	LUGAR	R
62	MURKOWSKI	R
63	DOMENICI	R
64	WARNER	R
65	HAGEL	R
66	MCCAIN	R
67	LOTT	R
68	COCHRAN	R
69	BENNETT	R
70	HATCH	R
71	BOND	R
72	MARTINEZ	R
73	ROBERTS	R
74	ALEXANDER	R
75	GRASSLEY	R
76	HUTCHISON	R
77	DOLE	R
78	SUNUNU	R
79	BROWNBACK	R
80	CORKER	R
81	CRAIG	R
82	SHELBY	R
83	CRAPO	R
84	BARASSO	R
85	GREGG	R
86	MCCONNELL	R
87	THUNE	R
88	ISAKSON	R
89	CHAMBLISS	R
90	GRAHAM	R
91	VITTER	R
92	CORNYN	R
93	SESSIONS	R
94	BUNNING	R
95	KYL	R
96	ENZI	R
97	BURR	R
98	ALLARD	R
99	ENSIGN	R
100	INHOFE	R
101	DEMINT	R
102	COBURN	R

Rank	Name	Party
1	MENENDEZ	D
2	LAUTENBERG	D
3	SCHUMER	D
4	DURBIN	D
5	CANTWELL	D
6	MURRAY	D
7	CARDIN	D
8	BROWN	D
9	WHITEHOUSE	D
10	BOXER	D
11	REED	D
12	KERRY	D
13	HARKIN	D
14	SANDERS	D
15	LEAHY	D
16	AKAKA	D
17	BINGAMAN	D
18	LEVIN	D
19	STABENOW	D
20	WYDEN	D
21	KOHL	D
22	FEINSTEIN	D
23	KLOBUCHAR	D
24	MIKULSKI	D
25	REID	D
26	CASEY	D
27	NELSON	D
28	FEINGOLD	D
29	SALAZAR	D
30	WEBB	D
31	KENNEDY	D
32	ROCKEFELLER	D
33	INOUYE	D
34	CONRAD	D
35	CARPER	D
36	DORGAN	D
37	TESTER	D
38	BAUCUS	D
39	LINCOLN	D
40	DODD	D
41	BIDEN	D
42	BYRD	D
43	LIEBERMAN	D
44	CLINTON	D
45	PRYOR	D
46	MCCASKILL	D
47	LANDRIEU	D
48	BAYH	D
49	OBAMA	D
50	NELSON	D
51	JOHNSON	D

Rank	Name	Party
52	SNOWE	R
53	COLLINS	R
54	SPECTER	R
55	SMITH	R
56	COLEMAN	R
57	WICKER	R
58	THOMAS	R
59	VOINOVICH	R
60	STEVENS	R
61	MURKOWSKI	R
62	LUGAR	R
63	MCCAIN	R
64	DOMENICI	R
65	WARNER	R
66	HAGEL	R
67	LOTT	R
68	COCHRAN	R
69	HATCH	R
70	ROBERT	R
71	BENNETT	R
72	MARTINEZ	R
73	ALEXANDER	R
74	BOND	R
75	GRASSLEY	R
76	BROWNBACK	R
77	HUTCHISON	R
78	CORKER	R
79	DOLE	R
80	SUNUNU	R
81	BARASSO	R
82	CRAIG	R
83	SHELBY	R
84	CRAPO	R
85	THUNE	R
86	ISAKSON	R
87	GREGG	R
88	MCCONNELL	R
89	CHAMBLISS	R
90	GRAHAM	R
91	SESSIONS	R
92	VITTER	R
93	CORNYN	R
94	BUNNING	R
95	ENZI	R
96	BURR	R
97	ALLARD	R
98	KYL	R
99	ENSIGN	R
100	INHOFE	R
101	COBURN	R
102	DEMINT	R

## 5.2 Using Internal Knowledge

We did another experiment where we picked as our two exemplars the senators who were the **least** similar in terms of their voting records. In this case our algorithm ended up picking Senator Robert Menendez of New Jersey and Senator Jim DeMint of South Carolina. This is interesting because while Senator DeMint has a reputation in the Senate as a staunch conservative, Senator Menendez does not have as high a public profile. This analysis suggests his voting record may be more partisan than his low profile reputation suggests. As per our methodology we fixed  $f_i = 1$  for Senator Menendez and  $f_i = -1$  for Senator DeMint and ran our algorithm and obtained the results shown on the previous page.

## 6 Discussion

First we note that the rankings produced are very reasonable and correlate well with rankings produced by interest groups and political commentators. The advantage of an data driven method of course is that it does not require human expertise. Secondly the “domain knowledge” and “internal knowledge” methods produce very similar results. This suggests a certain degree of robustness. The “internal evidence” method appears preferable as it does not require any choice of parameters.

### 6.1 Conclusions

We have presented a fast method for ranking legislators based on their votes. The method gives reasonable results, is easy to implement and apparently more straightforward than competing methods such as that of Poole. In addition the similarity matrix is an intuitive concept and suggests some applications in the area of visualizing the legislature.

### 6.2 Future Work

One interesting idea is to explore the idea of using different similarity functions. In this work we used the plain vanilla Hamming distance, about the simplest things that we could use. It is possible that a more sophisticated domain specific similarity function might produce qualitatively different result (e.g. diffusion kernel or Rank Similarity). Another idea is to further explore any significant qualitative differences with other ranking algorithms to establish the respective advantages and disadvantages of the various methods.

## References

- [1] S. Agarwal. Ranking on graph data. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006. (document), 1
- [2] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006. 1
- [3] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.
- [4] K. T. Poole. Voteview.com. <http://www.voteview.com>, 2008. 2.1, 5
- [5] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005. [http://www.cs.wisc.edu/~jerryzhu/pub/ssl\\_survey.pdf](http://www.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf).
- [6] X. Zhu. Semi-supervised learning with graphs. 2005. Doctoral Dissertation. 1, 2.2
- [7] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning*, pages 912–919, 2003. 1