Legislation approval ratings prediction via vote correlation

Yuan Deng, Junyang Gao, Xiaoxue Wang

Computer Science Department, Duke University

Objectives

We implement the machine learning techniques, including topic modeling, support vector clustering, and spectral partitioning, to predict the legislation approval ratings.

Our novelty is to incorporate a new feature called **vote correlation**, which represents the voting similarities / correlations between two voters.

Data Description

The data we used are crawled from open-source datasets from the web.

Vote	12348
Voter	955
Total Votes by voters	12592

Table 1: Dataset statistics

Outline of Algorithm

Different from the traditional way of prediction, in which based on the voting history of each voter, the system simply predicts the vote for the new legislation, our system works as follows:

1. For each pair of voters, compute vote correlations;
2. Classify voters into two groups by vote correlations;
3. Predict the aggregate vote of the two groups;

Generally, our algorithm incorporates following ingredients and techniques:

- Feature extraction for votes and bills
  Tech: (topic modeling [1]);
- Vote correlation computation
  Tech: (support vector clustering (SVC) [2]);
- Group partition
  Tech: (spectral partitioning of graphs [3, 4]);
- Aggregate voting
  Tech: (majority rules [5] and SVC);

Feature Extraction

To turn a legislation (a.k.a a bill) $b \in B$, to a law, U.S. government conducts several rounds of votes $v \in V_b$ for the corresponding bill $b \in B$.

We extract the feature of a vote by running topic modeling on the description of bills and votes separately to get feature vectors for bill $b \in B$ and vote $v \in V$ as F_b and F_v.

Finally, the feature vector of a specific vote $v \in V_b$ is the concatenation of F_b and F_v.

$$G_v = F_v \circ F_b$$

Compute Vote Correlation

For voter a and b, select the votes that both of them participated as $V_{a,b}$. Our objective is to train a model to predict whether voter a and b have the same opinion for a new vote.

We model the problem as a classification problem as follows: for each $v \in V_{a,b}$,

- let $x_v = G_v$;
- $y_v = 1$ if and only if a and b have the same vote for v; otherwise, $y_v = 0$;

We run a support vector clustering with radial basis function kernel to obtain a classifier.

Group Partitioning

Applying the classifier obtained by vote correlation computation, we can compute a matrix such that $M_{i,j} = 1$ if and only if we predict voter i and j have the same opinion for the considered vote.

In order to partition the group into two groups, we use *spectral partitioning of random graphs*

1. Compute singular value decomposition of M;
2. Recover the group members according to the top two eigenvectors;

Aggregate voting

After partitioning voters into two groups A and B, we treat each group as a super-voter (aggregate by majority rule) and predict its vote.

Again, we model the problem as a classification problem as follows: for each vote v

- let $x_v = G_v$;
- as for y_v:
  - if more than two thirds of voters vote for *yes*, then $y_v = 1$;
  - if more than two thirds of voters vote for *no*, then $y_v = 0$;
  - otherwise, discard v;

We run a support vector clustering with radial basis function kernel to obtain a classifier.

Experiments Setup

Due to computational complexity, it is impossible for us to run an experiment on the entire data sets. Thus, for each input of new vote, we select only 50 voters and predict the results among these 50 voters.

Results and Discussion

We use *leave-one-out cross-validation* to measure the performance of our algorithm and the correctness rate of our algorithm is roughly 72% and the precision / recall matrix is as follows:

	0	1
0	261	83
1	120	262

Table 2: Precision / recall matrix

The correct rate is acceptable but there is a huge space for further improvement. Several possible ways may include

- Use a better topic modeling to extract the features of votes;
- Use other model to train the classifier than SVC;
- In aggregate voting, maybe other rules than majority rule can be implemented;

Conclusion

We successfully incorporate several machine learning techniques to develop a new way for legislation approval ratings predictions. Different than the traditional method, we examine the second order information, vote correlation between pairs of voters to improve the performance of algorithms.

References

[1] Shawn Graham, Scott Weingart, and Ian Milligan. Getting started with topic modeling and mallet. *The Programming Historian*, 2, 2012.

[2] Asa Ben-Hur, David Horn, Hava T Siegelmann, and Vladimir Vapnik. Support vector clustering. *The Journal of Machine Learning Research*, 2:125–137, 2002.

[3] Frank McSherry. Spectral partitioning of random graphs. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 529–537. IEEE, 2001.

[4] Amit Kumar and Ravindran Kannan. Clustering with spectral norm and the k-means algorithm. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 299–308. IEEE, 2010.

[5] Anthony J McGann. The tyranny of the supermajority how majority rule protects minorities. *Journal of Theoretical Politics*, 16(1):53–77, 2004.

Contact Information

- Website:
  - `http://cs.duke.edu/~ericdy`
  - `http://cs.duke.edu/~jygao`
- Email:
  - ericdy@cs.duke.edu
  - jygao@cs.duke.edu
  - xxw211@cs.duke.edu

Duke UNIVERSITY

DUKE COMPUTER SCIENCE