Communities and polarization in signed networks: European Parliament voting dataset

Emma Fraxanet
(Dated: April, 2021)

I. INTRODUCTION: SIGNED NETWORKS

Signed networks can be interesting because relevant community structure can emerge from their study. They are also useful to study polarization in social networks, which is a growing topic and current issue. There is not a lot of ways to visualize this type of networks and their properties. There are methods to obtain the community structure that differ in their approach. However, the goal is the same for all of them: minimizing the proportion of frustrated edges, which are the negative edges within a community and the positive edges between two different communities. This is also expressed as the maximization of the structural balance [1] or edge-agreement ratio[2], which are the opposite measures to the frustrated edges ratio.

At the same time, the methods can differ in how they consider the relevance in the positive interactions, either by setting a threshold in building the network, or allowing the presence of a neutral group of nodes that doesn't belong to any detected community.

In this project we will compare and apply to our dataset three different methods: the first one uses an approximation of the Potts model based on a continuous convex relaxation using hinge-loss functions [1], the second maximizes the modularity finding a community structure in the agreeing part of the network using the Louvain method and, finally, the third one uses spectral algorithms based on a discrete eigenvectors problem [2]. The code used for such studies is extracted from the repository of each respective article and is assisted by the use of supporting code and *GEPHI*.

Some baseline algorithms using the spectral based algorithm structure are also compared.

The main research questions in this project are:

1. How do the methods differ in providing the maximum structural balance/edge-agreement ratio.
2. Do the methods provide similar communities.
3. How do the methods change with a different egde threshold.
4. Is the community structure derived from right/left or pro-european/anti-european antagonisms.

The last question follows some interesting results from reference [1], where it is clearly interesting to compare the community structure with the ground truth political parties of each member.

II. DATASET

The dataset studied in this project is a recompilation of 300 voting procedures in the European Parliament from May 2014 to June 2015, obtained from http://www.votewatch.eu/. This dataset and its posterior manipulation to obtain a network is extracted from the Santamaría,G., Gómez, V. [1] article and code repository.

The basic characteristics of the obtained network are:

1. Nodes represent members, there are only 730 members since only the ones that stayed through the whole term are considered. The information of the political party of each member is also available.

2. Edges represent agreement(disagreement) over all the votings between two members by having a signed weight +1 (−1).

The relevance of a voting event is measured with the discepancy value, which is based on the entropy (Eq.1) of the event. It follows, then, that a voting event where most members agree has less relevance.

$$H_r = -\frac{n_f^r}{n_{tot}^r} \log \frac{n_f^r}{n_{tot}^r} - \frac{n_a^r}{n_{tot}^r} \log \frac{n_a^r}{n_{tot}^r} \tag{1}$$

Where n_f^r is the number of *for* votes in the voting event r, n_a^r is the number of *against* votes and n_{tot}^r the total number of votes. Together with this measure and the agreement/disagreement between two members (v_i, v_j) in the same voting event r over all events one can obtain a spectrum of agreement/disagreement (Eq.2). The choice of whether to consider this interaction with the given sign will be based on using a threshold for these values. This threshold can filter interactions with low weights, and leads to a more or less sparse network.

$$\lambda(i,j)^r = \begin{cases} H_r \frac{n_a^r}{n_{\text{tot}}^{\text{ot}}}, & \text{if both voted for} \\ H_r \frac{n_f^r}{n_{\text{tot}}^r}, & \text{if both voted against} \\ -H_r \max\left(\frac{n_f^r}{n_{\text{tot}}^r}, \frac{n_a^r}{n_{\text{tot}}^r}\right), & \text{if voted different} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

By using different thresholds on the network we end up keeping three values: 0 (no threshold), 10 and 19. The network obtained with no threshold is much more dense. The network obtained with threshold 19 is mostly made of negative edges. This is good to keep in mind for the following results.

III. METHODS

A. POTTS MODEL

The method is build on a probabilistic inference problem with the approximation of the Potts model (used in statistical mechanics) based on a continuous convex relaxation using hinge-loss functions. It can also be seen as a Hinge-loss minimization. Generally, the idea is to define the problem as a set of soft and hard rules and find the state that minimizes the hinge-loss function, which is the distance to satisfaction of the soft rules with hard rules constraints. This state will define the community structure.

This model allows to choose the number of communities. In this project we focus on 2 and 3 communities because it is when the comparison with other methods is more feasible. Since the reference from this model [1] found threshold 19 to be the most optimal in structural balance, we use this threshold for our study.

B. MODULARITY CLASS WITH POSITIVE NETWORK

In this method we essentially obtain the communities that maximize modularity. Modularity (Eq.3) measures the density of links inside communities versus the links between communities. It is a measure done on not signed networks, so in this case we only keep the positive edges, which

Algorithm 1 EIGENSIGN

Input: adjacency matrix A

Compute $\mathbf{v}$, the eigenvector corresponding to the largest eigenvalue λ_1 of A.

Construct $\mathbf{x}$ as follows: for each $i \in \{1, \ldots, n\}$, $x_i = sgn(v_i)$.

Output $\mathbf{x}$.

Algorithm 2 RANDOM-EIGENSIGN

Input: adjacency matrix A

Compute $\mathbf{v}$, the eigenvector corresponding to the largest eigenvalue λ_1 of A.

Construct $\mathbf{x}$ as follows: for each $i \in \{1, \ldots, n\}$, run a Bernoulli experiment with success probability $|v_i|$. If it succeeds, then $x_i = sgn(v_i)$, otherwise $x_i = 0$.

Output $\mathbf{x}$.

Figure 1: Algorithm structure for the Eigensign and Random-Eigensign methods. Figure extracted from the reference [2]. Besides these structures there are two practical enhancements applied: for Eigensign an optimal threshold is calculated to obtain neutral nodes and for Random-Eigensign a $\|\mathrm{v}\|_1$ multiplicative factor is added to the probability.

means we only consider agreement in the community structure. The results are obtained using the Louvain method from the *GEPHI* software.

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta\left(c_i, c_j\right) \tag{3}$$

Since threshold 19 leaves a mostly negative network, we only apply this method to the network with no threshold and with threshold 10.

C. EIGENSIGN AND RANDOM EIGENSIGN

This methods are based on a discrete eigenvectors problem [2]. The idea is to find two polarized communities that maximize polarity (Eq. 4) while allowing neutral nodes. The expression for the polarity is penalized with the size of the vector, therefore only relevant nodes are added to the polarized communities. From a network $G = (V, E_+, E_-)$ the method tries to find a vector $x \in \{-1, 0, 1\}^n$ that maximizes P.

$$P = \frac{\mathrm{x}^T A \mathbf{x}}{\mathrm{x}^T \mathrm{x}} \tag{4}$$

For this methods we work with the three obtained networks: from no threshold, threshold 10 and threshold 19.

D. BASELINE METHODS

The following methods use the eigenvector components from the previous algorithms and serve as baseline algorithms. Therefore, they are based on two polarized communities with neutral nodes as well, and we also use the three threshold options. All information of these methods is extracted from reference [2], as well as the code to implement them.

- GREEDY: iterative method that removes the vertex that minimizes the difference between the number of positive and negative edges until finding an empty graph. Then returns the subgraph that had higher polarity and assigns those nodes to a cluster given their sign in the eigenvector. Based on *densest subgraph* from reference [4].
- BANSAL: This alghorithm identifies for each node two communities: one that disagrees with it and another one that agrees with it. The other nodes are neutral. From all the obtained communities for each node it returns the one with maximum polarity. Inspired by reference [5].
- LOCAL SEARCH: Local search approach guided by the polarity function. Starts with a random set of nodes and iteratively adds or removes nodes to the current situation according to the maximization of polarity. It ends when the gain in polarity is no larger than 0.2. It assigns the nodes to communities given their sign in the eigenvector.

IV. VISUALIZATION OF THE DATA

As mentioned before, the visualization of signed networks can be tricky. An option found for this visualization can be done with *GEPHI*. In Fig.2 we present the network separated by the sign of the edges. In this case we focus on the network obtained deleting the negative edges. We can see that, with the use of the Force Atlas 2.0 layout in the positive edges in Fig.2, we obtain a clustering that adapts nicely to the ground truth party structure, as well as other structures obtained with some of the methods mentioned.

We can also use this same technique for the network obtained with a threshold. In this case we only keep the connected components, as some nodes are isolated (the network becomes more sparse). See the results in Fig.3. We see that the clustering defines most parties (or groups of parties) clearly.

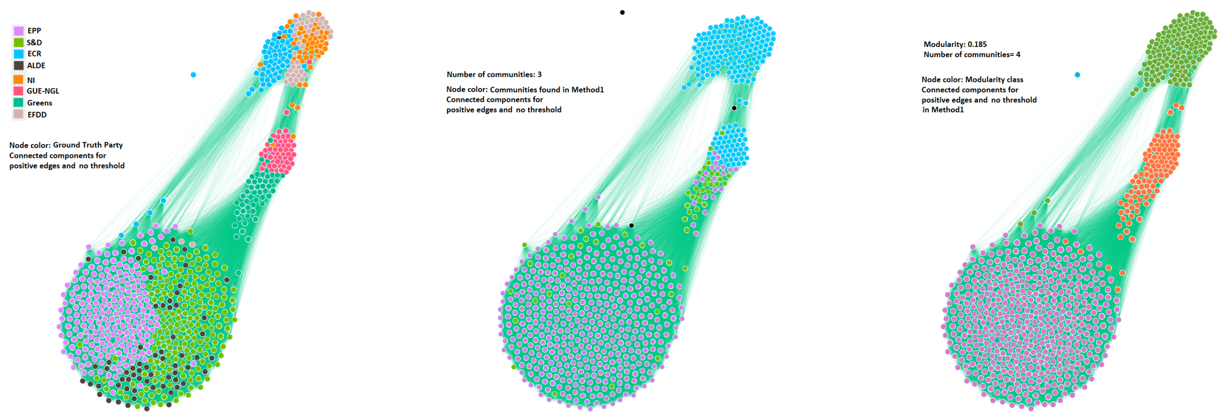

Figure 2: Visualization of the network with positive edges and the Force Atlas 2.0 layout from *Gephi*. The color code from the node belongs to the respective community structures of the ground truth parties, the potts model with k=3 and the modularity class.

V. RESULTS

A. FRUSTRATED EDGES RATIO: EFFICIENCY OF THE METHODS

To assess the efficiency of the methods we measure the ratio of frustrated edges over the whole network (without threshold) (Fig.4).

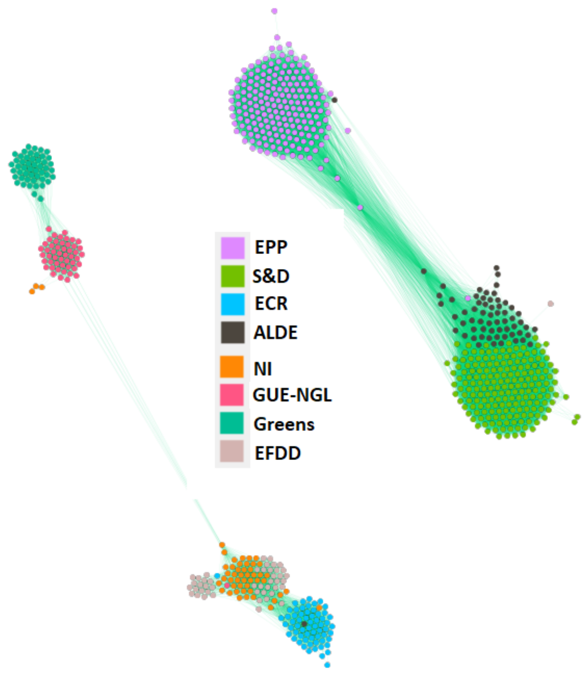

Figure 3: Visualization of the network with positive edges and the Force Atlas 2.0 layout from *Gephi* and threshold 10. The color code from the node belongs to the community structure of the ground truth parties.

Since the goal is to minimize this variable, the Modularity method with no threshold and the Eigensign/Random-Eigensign methods with no threshold are the ones that deliver the best performance. The fact that the methods that allow neutral nodes obtain more satisfactory results was expected. However, the Modularity method for threshold 0 produces surprisingly good results.

Note that in Fig.4 we don't present results for the last methods and higher thresholds. That is because the results weren't satisfactory. This means that in the following study of communities we will prefer those methods without threshold.

Overall, most methods presented in Fig.4 deliver good results, since the maximum value is around 0.25, which would mean that one fourth of the edges is being miss placed.

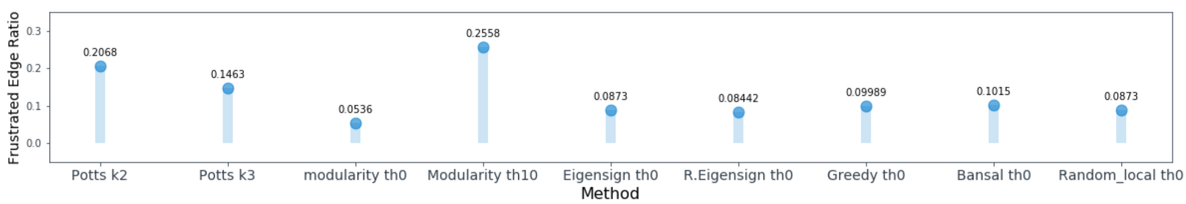

Figure 4: Frustrated edge ratio for each of the methods over the whole network.

B. COMMUNITY FINDINGS

In this section the community structure is presented. Both figures 5 and 6 are bar charts where each bar belongs to a member. The color and height of the bar is the community that the member is assigned to. The members are ordered and grouped by the ground truth political parties.

If we first focus on Fig.5 we note that using higher threshold for the Modularity method leads to finer tuning between the right-left spectrum. The other methods tend to separate communities agreeing with the pro-europeanism versus EU-scepticism antagonism. Even if using higher thresholds for the methods Eigensign and Random Eigensign delivered unefficient structural balance results, we find interesting that for a moderate threshold the Random-Eigensign switches the structure to a right versus center and left wing antagonism. The Eigensign with threshold 10 does a similar switch but the GUE-NGL party is considered completely neutral, probably because it falls too far left on the spectrum.

However, we also see the uncertainty when putting the Greens party in one community or the other in the Potts method or the neutral group in the case of Eigensign/R.Eigensign algorithms. For the 3-community Potts method it even has a community of its own. This makes sense because the Greens party is not comfortable with the more conservative/center parties from the Pro-Europeanism block but at the same time can not move to the EU-scepticism block.

For the case of the Modularity method, that is based only in agreements, the Greens party is clearly put in the Left-wing community with GUE-NGL.

In the baseline methods (Fig. 6) we see that for no threshold most methods return communities agreeing with the Pro-Europeanism versus EU-scepticism antagonism. The Local Search (Random Local) method detects the neutral behaviour of the Greens party. Even though using higher thresholds leaded to bad structural results we found interesting that using a moderate threshold switches again the structure to the right-left/center antagonism for the Greedy and Local Search methods.

Figure 5: Community structure for the main methods.

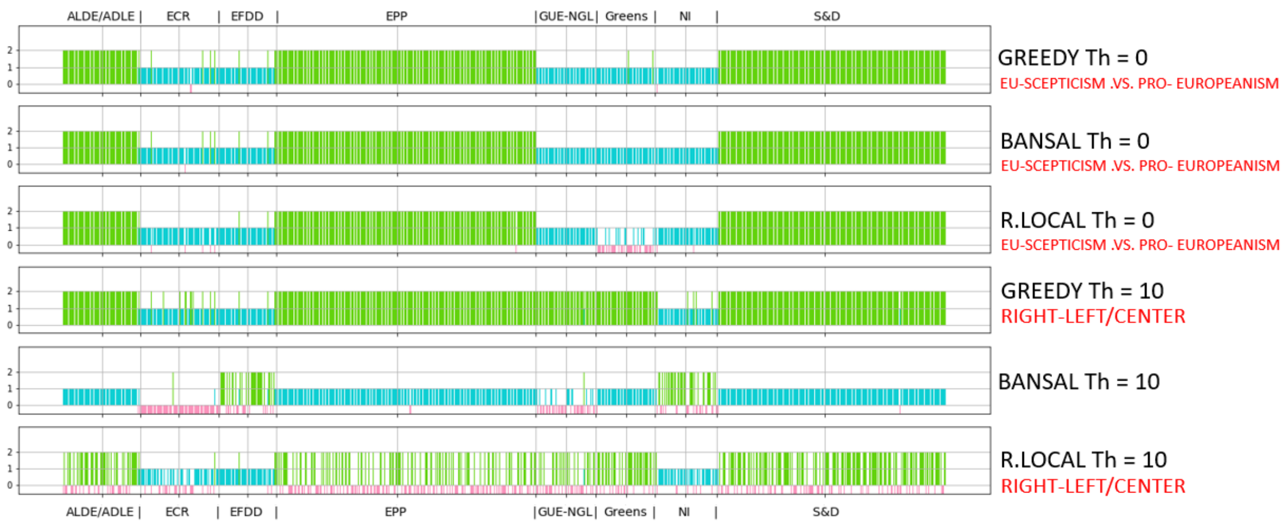

Figure 6: Community structure for the baseline methods.

VI. SUMMARY AND CONCLUSIONS

We found that the methods used were successful in detecting the different ground truth parties and, moreover, grouped them into meaningful groups. These groups in some cases considered the Pro-Europeanism versus EU-scepticism antagonism and in other cases had more to do with the right-left wing spectrum.

We found that allowing neutral nodes leaded to better structural balance, and minimized the frustrated edges ratio.

For the Potts method, which considered only disagreements, we found acceptable structural balance results and it maintained the Pro-Europeanism versus EU-scepticism structure for two and three communities. The fact that for three communities one community is formed by only members of the Greens party can be compared to the neutral group for the Eigensign method, which would mean that, in some way, the Potts method considers neutral nodes as a separated community.

For the Modularity method, which clustered members only according to the agreement, we found surprisingly good structural balance results for no threshold. For the community structure we found that it was based on the right-center-left spectrum. Adding a moderate threshold, though with a larger proportion of frustrated edges, delivered a finer tuning on the detection of this spectrum.

The fact that these two mentioned methods (Potts and Modularity) obtain different structures and are based mostly on negative/positive edges leads to the conclusion that it is interesting to study the disagreement and agreement interactions separately to obtain emerging communities.

For the other methods, conisdering both Eigensign, Random-Eigensing and the baseline methods we find satisfactory structural balance results when we have no threshold. We also find that with no threshold the communities resemble the Pro-Europeanism versus EU-scepticism, while the communities with threshold 10 follow more the right-left wing spectrum.

For future work we leave the idea of separating votings by topic, to see in which topìcs is one the two different main structures more relevant.

VII. REFERENCES

[1] Santamaría, G., Gómez, V. (2015). Convex inference for community discovery in signed networks.

[2] Bonchi, F., Galimberti, E., Gionis, A., Ordozgoiti, B., Ruffo, G. (2019, November). Discovering polarized communities in signed networks. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management (pp. 961-970).

[3] M. Charikar. Greedy approximation algorithms for finding dense components in a graph. In International Workshop on Approximation Algorithms for Combinatorial Optimization, pages 84–95, 2000.

[4]N. Bansal, A. Blum, and S. Chawla. Correlation clustering. Machine learning, 56(1-3):89–113, 2004.