Analyzing Voting Data Using Graphical Clustering Methods

2020 Fall MATH 285J
Minimal surface and curvature driven models in Image Processing and Machine Learning

Kyung Ha, Grace Li, Jiayi Li
Blaine Talbut, Thomas Tu

University of California, Los Angeles

December 18, 2020

1 Introduction

In this investigation, we utilize voting records in conjunction with clustering and community detection algorithms to classify legislators into communities by political stance. The underlying assumption is that legislators with more similar voting records have more similar political stances. We consider legislatures from multiple countries: the United States House of Representatives, German Bundestag, Legislative Council of Hong Kong, and South Korean National Assembly. For each legislature, we collect roll call voting data and apply five different similarity functions to construct similarity matrices of the legislators (section 2). We then apply spectral clustering (sections 3 and 4), Louvain with and without k-nearest neighbors preprocessing (sections 4 and 5), and MBO modularity maximization (section 6) methods to the similarity matrices.

For each legislature, we assess how well the combinations of similarity matrices and algorithms can detect existing, known political parties. We find that for legislators that are registered as independents with no political-party affiliation, we are able to identify what larger political party coalitions the legislators are most aligned with. In addition, by using multi-class methods, we determine if subclusters of political factions can be detected. By placing time periods where the legislatures had greater number of clusters in historical context, we can demonstrate that our methods can be used as an indication of unity or division within a political party, and of when there may be a shift in power within the legislature.

2 Grace Li: U.S. Data and Similarity Matrices

2.1 U.S. House of Representatives Data

The roll call voting data for the United States Congress is curated and made publicly available by the voteview project [21]. Because the House of Representative and Senate have different votes, we decided to only focus on one chamber and chose the House of Representatives due to its larger membership, which could lead to more interesting clustering. From the voteview database, we collected the roll call voting data for the 57th to 115th Houses, which ranged from 1901 to 2019. The current 116th House was not considered since it is still in session and its votes are incomplete. Each Congress had some changes in membership during its two-year period such as losing members due to resignation or moving to a cabinet position, and gaining members due to governor appointment or special election. In our analysis of the U.S. House, we only consider representatives who participated in at least 25% of the roll call votes. The voteview data set provides the political-party affiliation of each legislator. These affiliations are predominantly democrat or republican, but there are also a handful of independents and other party affiliations, including the progressive and farmer-labor parties, particularly in earlier Congresses of the 20th century. For each House, we constructed a roll call matrix with each column representing a legislator and each row representing a roll call vote. The matrix was populated with vote values of 1 for yes, -1 for no, and 0 for abstention or no data available.

2.2 Fraction Similarity

The similarity matrices we construct in order to cluster the U.S. House and the legislatures of other countries are inspired by prior work on the U.S. Congress. In their paper, Waugh et al. [27] construct similarity matrices using the fraction of votes that were in agreement once any abstentions were ignored. This is based on the assumption that legislators who vote the same way should be more politically aligned. The fraction similarity function for legislators X and Y is given as follows

$$S_{fraction}(X, Y) = \frac{\text{number of votes } X \text{ and } Y \text{ voted in agreement}}{\text{number of votes } X \text{ and } Y \text{ both participated in}} \tag{1}$$

Waugh et al. [27] calculate the fraction-similarity matrix for the 1st through 109th House and Senate, ranging from 1789 to 2007. Using a variety of heuristic algorithms, they detect communities of legislators by maximizing modularity Q, which is defined as

$$Q = \frac{1}{2m} \sum_{1 \leq i,j \leq n} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(g_i, g_j) \tag{2}$$

where A is the similarity matrix, n is the number of nodes in the network, $k_i = \sum_{j=1}^{n} A_{ij}$ is the weighted degree of node i, $m = \frac{1}{2} \sum_{i=1}^{n} k_i$ is the total strength of ties in the network, g_i is which community node i is assigned to, and $\delta(g_i, g_j)$ is 1 if $g_i = g_j$ and 0 otherwise. When modularity is higher, then tie strengths are more concentrated within the partitioned communities and there is greater clustering. In addition to heuristically determining the maximum-modularity partition, Waugh et al. also calculated P, the modularity from splitting the Congress along party lines. They demonstrate that the ratio P/Q indicates the extend to which polarization is partisan, and in years with high P/Q, polarization is predominantly partisan. In years where P/Q is low, divisions other than party lines better explain polarization, and that may be an indication of party reorganization or a shift in the majority power [27]. We use the modularity function to cluster legislators in section 6.

2.3 Rajski Similarity

Another similarity matrix we construct is based on work by Jakulin et al. [20] that analyzes the first session of the 108th Senate in 2003 and utilizes ideas from information theory. The authors note that if we ignore abstentions, then the binary yes or no voting options are consistent with the binary model of Shannon's theory of information. For any ordered pair of senators (X, Y), we define $p_{i,j}$ to be the fraction votes in which senator X voted i and senator Y voted j out of the total number of votes for which senators X and Y both participated in the vote, where $i, j \in \{\text{no, yes}\}$. If we view each senator as a binary random variable which can vote yes or no, then if the votes of the two senators are dependent, the probability of vote outcome (i, j) is $p_{i,j}$. However, if the votes of the two senators are independent, then the probability of outcome (i, j) is $p_{i,*}p_{*,j} = (p_{i,i} + p_{i,j})(p_{i,j} + p_{j,j})$, where $p_{i,*}$ is the fraction of votes for which X voted i, regardless of what Y voted, and $p_{*,j}$ is the fraction of votes for which Y voted j, regardless of what X voted. Shannon's entropy is a measure of the average outcome uncertainty, or, equivalently, the amount of information conveyed by identifying the outcome of a

random variable. For the dependent and independent voting models, the entropy is given by

$$\text{Dependent Model: } H(X,Y) = -\sum_i \sum_j p_{i,j} \log_2(p_{i,j}) \tag{3}$$

$$\text{Independent Model: } H(X) + H(Y) = -\sum_i p_{i,*} \log_2(p_{i,*}) - \sum_j p_{*,j} \log_2(p_{*,j}) \tag{4}$$

The mutual information $I(X;Y)$ for the two senators X and Y is a measure of the reduction in uncertainty of the vote of one senator given the vote of the other. It is calculated by taking the difference in entropy of the independent and dependent models.

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{5}$$

Rajski's distance is a way of converting mutual information into a metric and is given by $d(X,Y) = 1 - \frac{I(X;Y)}{H(X,Y)} \in [0,1]$. Jakulin et al. utilize Rajski's distance to construct a dissimilarity matrix of senators, to which they apply agglomerative hierarchical clustering algorithms [20]. Using the Rajski's distance and the clustering algorithms, they clustered the legislators into political parties and found hierarchical subclusters of moderates and other party divisions. However, we are working with algorithms that are based on similarity matrices. Hence, we convert the Rajski's distance to a similarity function as follows

$$S_{Rajski}(X,Y) = 1 - d(X,Y) = \frac{I(X;Y)}{H(X,Y)} \in [0,1] \tag{6}$$

One feature of the Rajski similarity function is that due to the nature of mutual information, a pair of legislators will have the maximum similarity of 1 if they either always vote in agreement or always vote in disagreement. This may seem at odds with the assumption that legislators who vote together have similar political views. However, as Jakulin et al. point out, there are always votes which pass or fail with near majority, and the situation of two legislators voting in near complete disagreement over the course of the Congress does not occur [20]. Although the Rajski functions have this feature, Jauklin et al. were still able to get good clustering from their dissimilarity matrix. As we will see in later sections, the Rajski similarity performs well with the algorithms we apply.

2.4 SVD Similarity

The fraction and Rajski similarities only consider the votes that legislators participated in. However, abstentions, particularly those actively decided by the legislator, can also

convey information about the legislator. The remaining similarity matrices presented consider the entire roll call matrix with abstentions of value 0. For the U.S. Congress, the NOMIANTE scoring system developed by political scientists Keith Poole and Howard Rosenthal showed that 2-dimensions is a sufficient representation of how legislators vote [15]. In their paper, Porter et al. [25] apply SVD to the roll call matrices for the U.S. House and Senate and represent each legislator's voting record in 2-dimensions by only keeping the coordinates associated with the two principal singular values. By considering the sum of the square of the errors, we get that the kth singular value σ_k accounts for $\sigma_k^2 / \sum_{i=1}^{n} \sigma_i^2$ of the voting matrix. For the most recent 115th House from 2017–2019, the first singular-value coordinate accounts for 57.9% of the roll call matrix, the second coordinate accounts for 24.2% of the matrix, and each subsequent coordinate accounts for less than 2.1% of the matrix. Similar percentages are observed for the other Congresses, and this suggests that two coordinates gives a reasonable approximation of the roll call matrix. One benefit of the 2-coordinate representation is that it becomes easy to visualize. Figure 1 shows plots of the SVD coordinates of legislators colored by party-affiliation.

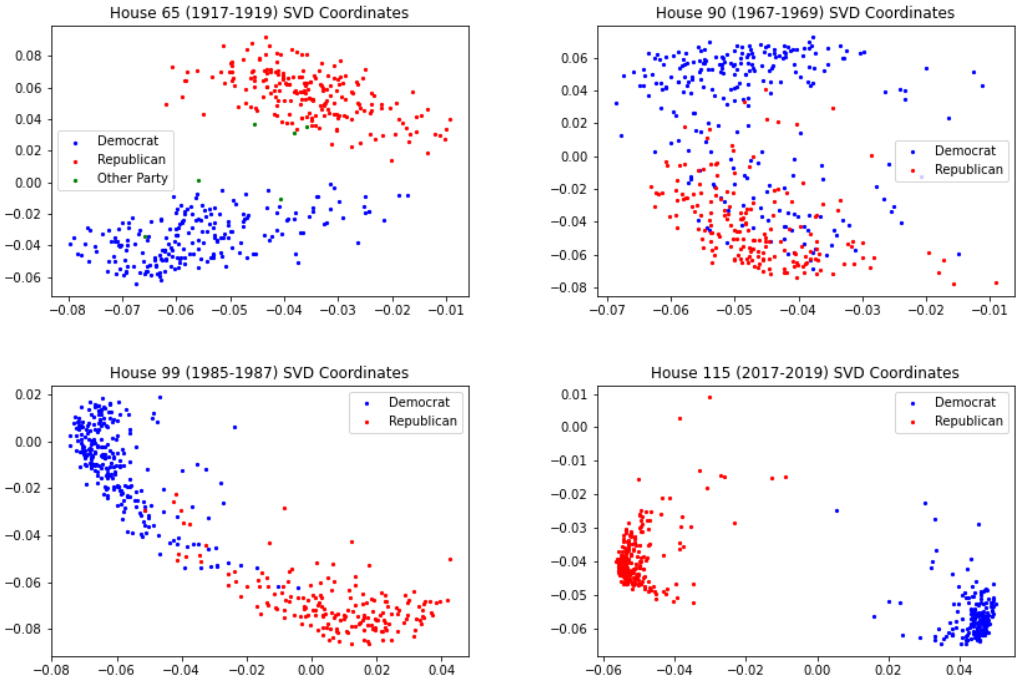


Figure 1: Sample plots of House legislators represented with SVD coordinates

From Figure 1, we see that the one coordinate is an indicator of political-party affiliation. For the 65th House (1917-1919) and 115th House (2017-2019) and other recent Houses, the partisan split in the SVD coordinates is clearly visible. However, this is not always the case. In the 99th (1985-1987) House, the middle of the plot has some democrats and republicans mixing, and in the 90th House (1967-1969) there is little separation between parties. For our clustering investigation, we apply a similarity function in the form of a Gaussian kernel to the 2-dimensional SVD coordinates of each legislator. The similarity between legislators X and Y with SVD coordinates x and y is defined as

$$S_{SVD}(X,Y) = \exp\left(\frac{-||x-y||_2^2}{2\sigma^2}\right) \tag{7}$$

where we take the scaling weight to be $\sigma = 0.1$ to keep the same levels of scaling as other similarity functions.

2.5 Local Similarity

Finally, we apply two additional similarity functions that are not specific to the roll call voting problem. In their paper, Zelnik-Manor and Perona [28] propose a local, self-tuned similarity function for use with spectral-clustering algorithms. Their similarity function is designed to improve clustering of points with different densities. The motivation for trying this function was that it might aid picking out subclusters of moderate legislators of either political party that visually appeared in areas that were less dense in the SVD plots. The local-similarity function for two legislators given their entire roll-call-voting vectors X and Y is defined as follows

$$S_{local}(X,Y) = \exp\left(\frac{-d(X,Y)}{\sigma_X \sigma_Y}\right), \text{ where } \sigma_X = d(X,X_k) \tag{8}$$

with X_k denoting the kth nearest neighbor of X. We take the distance function d to be the Euclidean distance $d(X,Y) = ||X-Y||_2$, and decided on using $k = 7$ after finding that experiments varying k did not effect the similarity matrix much for the U.S. data.

2.6 Cosine Similarity

The last similarity function that we will apply to the roll call matrices is the cosine similarity. The motivation for choosing this function is that the cosine function captures

whether two vectors are similar with an angle near 0°, unrelated with an angle near 90°, or opposite with an angle near 180°. Since the roll call votes include 0 values for abstentions, we hypothesized that these angle distinctions would be successful in picking up voter similarity. We use the form of the cosine-similarity function used by Hu et al. [18] which, given two entire roll call voting vectors X and Y with angle θ between them, is defined as follows

$$S_{cosine}(X,Y) = \exp\left(-\frac{\left(1-\frac{\langle X,Y\rangle}{||X||||Y||}\right)^2}{2\sigma^2}\right) = \exp\left(-\frac{(1-\cos\theta)^2}{2\sigma^2}\right) \tag{9}$$

We take the value of the weight parameter to be $\sigma = 1$ to keep the scaling of the similarity matrix in line with the other similarity matrices.

2.7 Summary and Discussion

To summarize the setup for our analysis, for each legislature, we first construct a roll call matrix with the columns representing legislators and the rows representing roll call votes. The votes are given values of -1, 0, and 1 for votes of no, abstain or no data, and yes respectively. Then, we explore five different types similarity matrices. The fraction (Eq. 1) and Rajski (Eq. 6) similarity matrices ignore abstentions and the similarity of two legislators only depends on votes they both participated in. The SVD (Eq. 7), local (Eq. 8), and cosine (Eq. 9) matrices utilize the entire roll call vote vectors of legislators including the abstention values. The SVD similarity first represents each legislator's roll call vote vector in 2-dimensions before applying a Gaussian kernel function, while the local and cosine similarity functions are directly applied to the entire roll call voting vector. In subsequent sections, we apply different clustering methods to these similarity matrices for various legislatures including the U.S. House, German Bundestag, Legislative Council of Hong Kong, and South Korean National Assembly.

For future work, we note that the SVD similarity function we applied was limited to keeping 2-dimensional coordinates. While we showed that this was a reasonable approximation for the U.S. legislature, this is not the case for other legislatures explored, such as the German Bundestag. However, as we will show in the proceeding sections, the SVD similarity was not as promising as the other similarity matrices. Hence, we did not focus our efforts on improving the SVD similarity. In any future work with the SVD similarity, different similarity functions other than the standard Gaussian kernel could

be applied to the reduced SVD coordinates. In addition different thresholds could be explored to determine how many principal singular values should be used to get the reduced SVD coordinates. A threshold could be set for what percentage of the matrix needs to be represented by the singular values when looking at the sum of the squares of the errors. In addition, future work on the similarity functions used in this paper could further tune the weight parameters to perhaps give faster or more accurate clustering depending on the algorithm used.

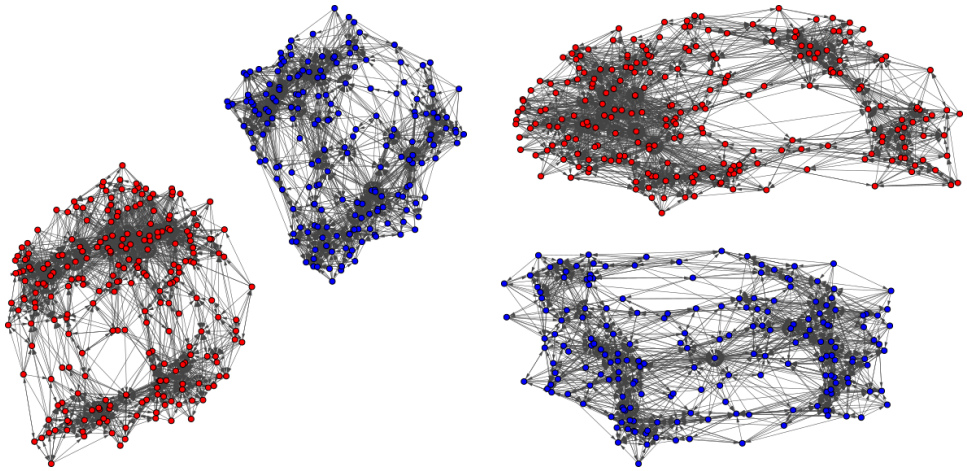
3 Jiayi Li: Spectral Clustering

3.1 Similarity Graphs

Given the data points $x_1, \ldots, x_n$ and pairwise similarities s_{ij}, there are different constructions to transfer it to a graph.

The ϵ-neighborhood graph. We connect all points whose pairwise distances are smaller than ϵ, where ϵ is a tuning parameter.

k-nearest neighbor graph. Given two distinct vertices v_i and v_j, the goal is to connect v_i and v_j if one is among the k nearest neighbors of the other. This leads to two ways in constructing the similarity graph: The first way is that we build an edge between v_i and v_j if and only if v_j is among the k nearest neighbors of v_i and v_i is among the k nearest neighbors of v_j; An alternative way is to connect v_i and v_j if either v_j is among the k nearest neighbors of v_i or v_i is among the k nearest neighbors of v_j. The first construction leads to a symmetric graph which is often referred to as the *mutual k-nearest neighbor graph*. The second construction results in a non-symmetric graph, which is what commonly referred to as the *k-nearest neighbor graph*.

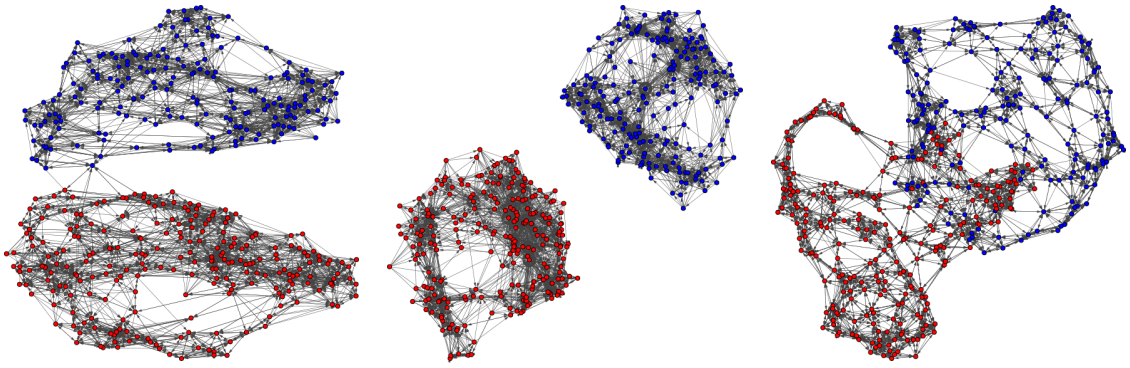


115th House legislators with cosine similarity

115th House legislators with fraction similarity

Taking the voting records of legislators from House 57-115, we construct the k-nearest neighbor graph for the similarity measures mentioned in the previous Chapter.

In the similarity graphs shown below, the red nodes represent the legislators who voted in the 115th House and were registered as republicans and the blue nodes represent those registered as republicans represented in the unsymmetrical knn graph with $k = 10$.



Local Gaussian similarity Rajski similarity SVD similarity

Note that k is a parameter to be selected based on the performance of the algorithm and different similarity function. In 57th-115th House legislators that we studied, $k = 10$ gives a relatively clear separation of the data nodes. In particular, for House 115, which corresponds to 2017-2019 (last two weeks of B.Obama and first two years of D. Trump), the cosine similarity measure, fraction similarity measure and Rajski similarity measure clearly separates the legislators based on their party affiliation.

3.2 Unnormalized Spectral Clustering

Using one of the five similarity measures on voting data described in Section 2, we construct a similarity matrix $S \in \mathbb{R}^{n \times n}$ with number of k clusters. We proceed to construct a similarity graph using knn and let W be the weighted adjacency matrix. We then compute the **unnormalized** Laplacian $L = D - W$ and the first k eigenvectors $u_1, \ldots, u_k$ of L. Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing $u_1, \ldots, u_k$ as columns. For $i = 1, 2, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U. Cluster the points $(y_i)_{i=1,\ldots,n}$ with the $k - means$ algorithm into clusters $C_1, \ldots, C_k$. For the algorithm, we are going to output clusters $A_1, \ldots, A_k$ with $A_i = \{j | y_j \in C_i\}$.

3.3 Normalized Spectral Clustering

There are slightly different spectral clustering algorithms that are referred to as normalized spectral clustering. Here we are using the algorithm that's proposed by Ng et al. in 2002 where we normalize the Laplacian matrix by the degree matrix. Using one of the five similarity measures on voting data described in Section 2, we construct a similarity matrix $S \in \mathbb{R}^{n \times n}$ with number of k clusters. We proceed to construct a similarity graph using knn and let W be the weighted adjacency matrix. We then compute the normalized Laplacian $\mathbf{L_{sym}} := \mathbf{D^{-1/2}LD^{-1/2}}$ and the first k eigenvectors $u_1, \ldots, u_k$ of L. Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing $u_1, \ldots, u_k$ as columns. Form the matrix T by normalizing the row sums of U to norm 1. For $i = 1, 2, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of T. Cluster the points $(y_i)_{i=1,\ldots,n}$ with the $k - means$ algorithm into clusters $C_1, \ldots, C_k$. For the algorithm, we are going to output clusters $A_1, \ldots, A_k$ with $A_i = \{j | y_j \in C_i\}$.

3.4 Cheeger Cut

Let $P = \{S, S^C\}$ be a partition of a graph F. The Cheeger cut value is defined as:

$$\mathcal{C}(S, S^C) = \frac{Cut(S, S^C)}{\min\{vol(S), vol(S^c)\}}$$

where

$$Cut(A, B) = \sum_{i \in A, j \in B} W_{ij}$$

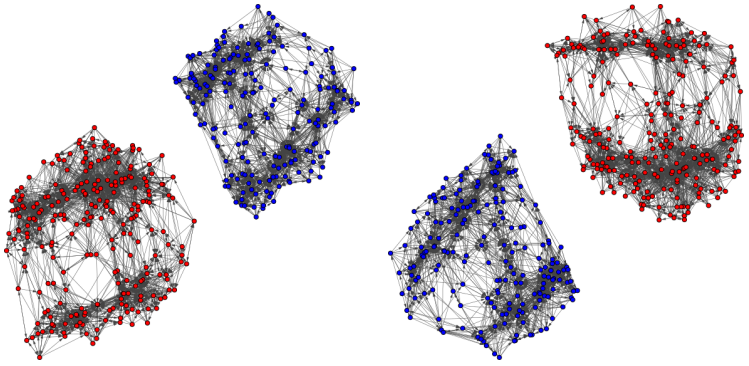
Let $\mathcal{C}_*$ be the Cheeger constant associated to G and λ_2 be the second smallest eigenvalue. The following inequality holds:

$$\frac{\mathcal{C}_*^2}{2 \max_i D_{ii}} \leq \lambda_2 \leq 2\mathcal{C}_*$$

3.5 Clustering Results

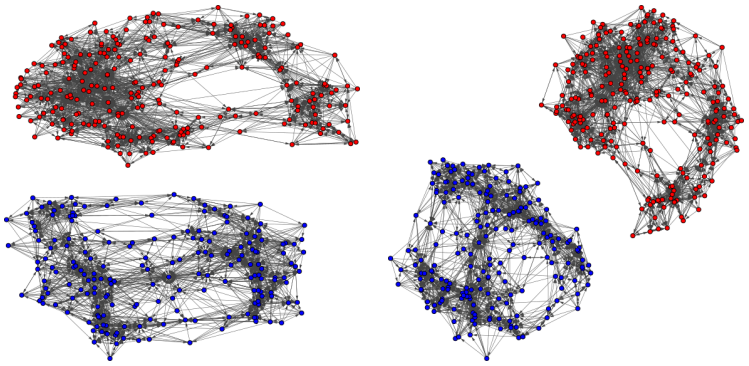
In this section, we are going to compare the clusters we obtain to the knn graph we obtained for each similarity measure. In each row, the graph on the left represents

the unsymmetrical knn graph with $k = 10$ for the labeled similarity measure. The graph on the right shows the party classification after applying the clustering algorithm. Red nodes represent legislators who are registered and correctly classified as Republican. Pink nodes represent legislators who are registered as Republican, but classified as Democrat. Navy nodes represent legislators who are registered and correctly classified as Democrat. Light blue nodes represent legislators who are registered as Democrat, but classified as Republican. The accuracy of the clustering is labeled at the bottom of each figure.



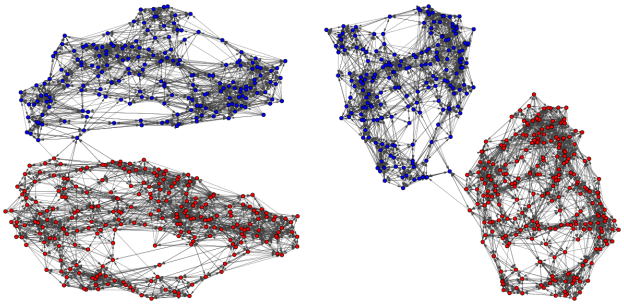
(a) knn graph (b) spectral clustering result

115th House legislators with Cosine Similarity, Accuracy = 1.0



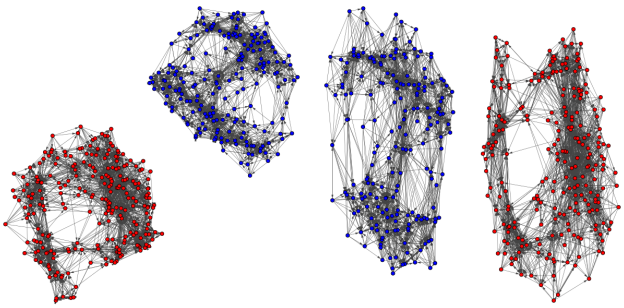
(a) knn graph (b) spectral clustering result

115th House legislators with Fraction Similarity, Accuracy = 1.0



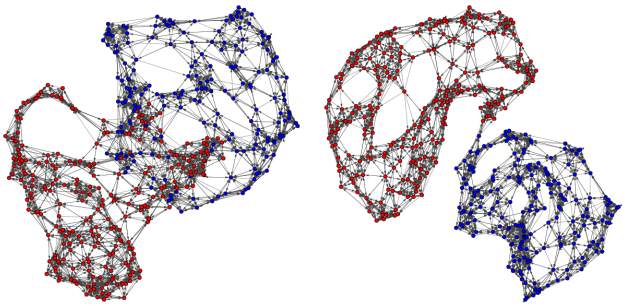
(a) knn graph (b) spectral clustering result

115th House legislators with Local Gaussian Similarity Similarity, Accuracy = 1.0



(a) knn graph (b) spectral clustering result

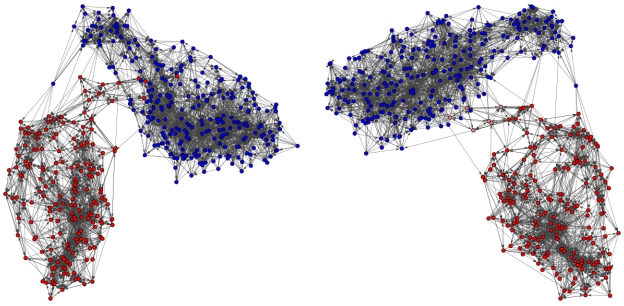
115th House legislators with Rajski Similarity Similarity, Accuracy = 1.0



(a) knn graph (b) spectral clustering result

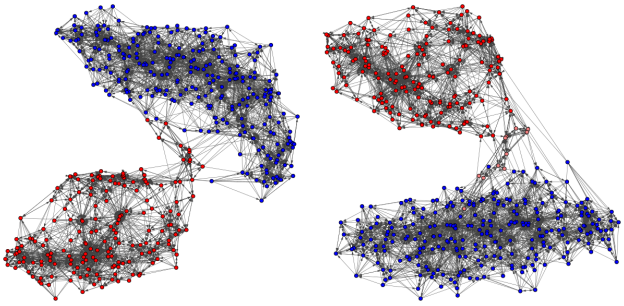
115th House legislators with SVD Similarity Similarity, Accuracy = 1.0

Summary and Discussion. For all the similarity matrices and spectral clustering methods that we worked with, the algorithm achieves a high accuracy (~ 0.98) on classifying the legislators to their registered party affiliation, indicating a high polarization among the US political parties. In particular, for House 110-115 (2007-2019), the algorithms achieves 1.0% classification accuracy for all the similarity measures considered. This alerts that the polarization among the US political parties has been worse over the recent years.



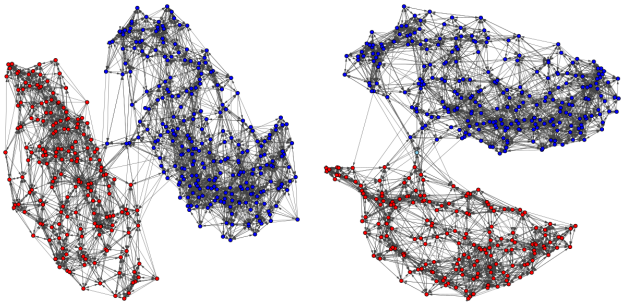
(a) knn graph (b) spectral clustering result

100th House legislators with Cosine Similarity, Accuracy = 0.986



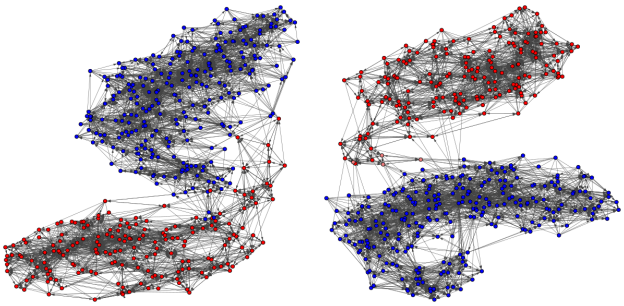
(a) knn graph (b) spectral clustering result

100th House legislators with Fraction Similarity, Accuracy = 0.975



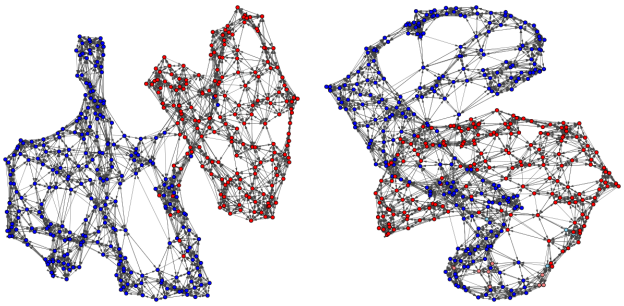
(a) knn graph (b) spectral clustering result

100th House legislators with Local Gaussian Similarity Similarity, Accuracy = 0.970



(a) knn graph (b) spectral clustering result

100th House legislators with Rajski Similarity Similarity, Accuracy = 0.991



(a) knn graph (b) spectral clustering result

100th House legislators with SVD Similarity Similarity, Accuracy = 0.970

4 Blaine Talbut: Multiclass Clustering

4.1 Louvain clustering algorithm

The Louvain algorithm [14] is a heuristic procedure for finding the cluster of a graph with maximum graph modularity:

$$Q(g) := \frac{1}{2m} \sum_{i,j=1}^{N} (w_{ij} - \gamma \frac{k_i k_j}{2m}) \delta(g_i, g_j) \tag{10}$$

where

- k_i is the weighted degree of node i,
- m is the number of edges,
- g_i is the community that node i belongs to.

The Louvain algorithm takes a hierarchical, locally greedy approach:

- Initially each node is in its own cluster.
- For each node, check whether modularity can be increased by adding it to a neighbor's cluster.
- Repeat until modularity cannot be improved in this manner.
- Contract each cluster to a single node.
- Repeat from step 1.

Note that, unlike with spectral clustering, it is neither possible nor necessary to specify the number of clusters in advance.

The effect on the modularity (10) of moving a node i from community C_1 to C_2 can be easily computed:

$$\Delta Q = \sum_{j \in C_2 \setminus \{i\}} \frac{w_{ij}}{m} - \sum_{j \in C_1 \setminus \{i\}} \frac{w_{ij}}{m} + \gamma \sum_{j \in C_1 \setminus \{i\}} \frac{k_i k_j}{2m^2} - \gamma \sum_{j \in C_2 \setminus \{i\}} \frac{k_i k_j}{2m^2}$$

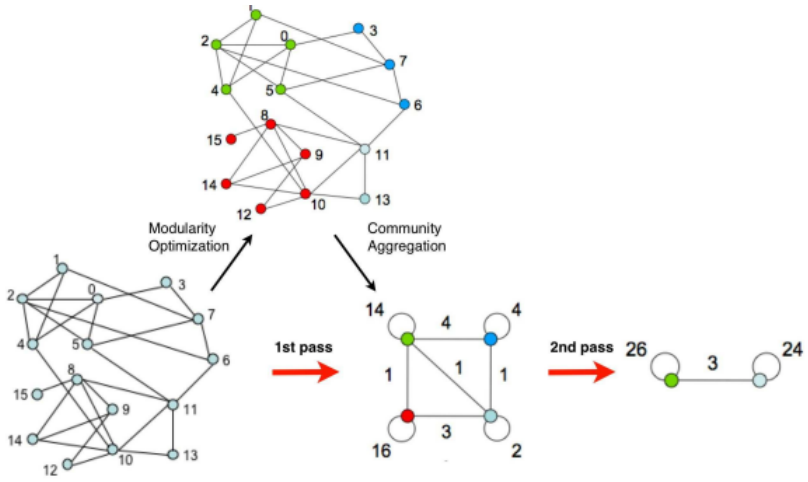


Figure 14: An illustration of two passes of the Louvain algorithm. Image from [14].

Figure **??** illustrates the method.

Compared to other heuristics for maximizing modularity, such as Graph MBO, Louvain is simple to implement.

4.2 A multiclass look at the US of Representatives

In [27], a small number of sessions of the US House of Representatives were found to cluster into more than two communities. These can be grouped as follows: Congress

Panel A: House

Congress	Community 1			Community 2			Community 3			Community 4		
	Size	*Divisiveness*	*Solidarity*	*Size*	*Divisiveness*	*Solidarity*	*Size*	*Divisiveness*	*Solidarity*	*Size*	*Divisiveness*	*Solidarity*
2	33 (46.5)	0.01038 (0.00246)	0.75 (0.214)	32 (45.1)	0.01019 (0.00244)	0.79 (0.216)	5 (7)	0.00879 (0.0013)	0.93 (0.057)	1 (1.4)	0.00533	1
9	67 (45.6)	0.00445 (0.00093)	0.62 (0.193)	45 (30.6)	0.00499 (0.00181)	0.58 (0.2)	35 (23.8)	0.00547 (0.00089)	0.79 (0.211)			
14	80 (41)	0.00418 (0.00091)	0.5 (0.179)	71 (36.4)	0.00445 (0.0011)	0.58 (0.207)	44 (22.6)	0.00429 (0.00096)	0.57 (0.168)			
15	96 (49.2)	0.00416 (0.00072)	0.52 (0.141)	96 (49.2)	0.00439 (0.00115)	0.48 (0.172)	3 (1.5)	0.00349 (0.00034)	0.68 (0.093)			
17	78 (39.2)	0.00382 (0.00087)	0.39 (0.158)	65 (32.7)	0.00374 (0.00101)	0.4 (0.162)	56 (28.1)	0.00404 (0.00107)	0.49 (0.213)			
21	110 (50)	0.00333 (0.00071)	0.64 (0.283)	109 (49.5)	0.00317 (0.00079)	0.69 (0.254)	1 (0.5)	0.00222	1			
32	100 (41.8)	0.00287 (5e-04)	0.57 (0.189)	87 (36.4)	0.00293 (0.00075)	0.57 (0.214)	52 (21.8)	0.00271 (0.00044)	0.65 (0.194)			
52	160 (47.1)	0.00201 (5e-04)	0.6 (0.214)	91 (26.8)	0.00255 (0.00055)	0.78 (0.159)	89 (26.2)	0.00202 (0.00047)	0.61 (0.158)			
81	172 (38.7)	0.00172 (0.00033)	0.79 (0.159)	169 (38.1)	0.00155 (0.00037)	0.82 (0.181)	103 (23.2)	0.00161 (0.00028)	0.79 (0.095)			
85	194 (43.7)	0.00167 (0.00044)	0.66 (0.177)	180 (40.5)	0.00157 (0.00028)	0.71 (0.216)	70 (15.8)	0.00168 (0.00022)	0.79 (0.118)			
86	211 (47.6)	0.00142 (3e-04)	0.75 (0.183)	161 (36.3)	0.00184 (0.00036)	0.75 (0.193)	71 (16)	0.00163 (0.00032)	0.78 (0.165)			

Figure 15: Sessions of the US Congress in which [27] found more than two communities. Image from [27].

2 is from an early period before party organization in the Congress. Congresses 9, 14, 15, and 17 belong to the period after the collapse of the Federalist party when the Democratic-Republicans exerted overwhelming control over Congress; here, intra-party factions were likely as or more salient than inter-party factions. Congress 21 belongs to the Jacksonian period directly precipitating the schism of the Democratic-Republicans

into the Democrat and Whig parties. Congress 32 appears on the eve of the collapse of the Whig party and the consolidation of the Whigs and other parties into the Republican party. Congress 52 was one of the few late–nineteenth century Houses controlled by the Democrats, and at that time the Democrats were split into an eastern pro-business faction and an agrarian populist faction. Congresses 81, 85, and 86 belong to the post-WW2 period where the Republican party was divided between an eastern liberal faction and an ascendant conservative faction.

From among these, only Congresses 81, 85, and 86 belong to our dataset [21].

To identify these clusters using our methods, there are a few parameters we can tune:

- the choice of metric used to generate the similarity graph (fraction, cosine, Rajsi, or SVD),
- whether and for which k to prune the similarity matrix using k-nearest neighbors,
- and the clustering algorithm to use (spectral clustering with k-means or Louvain).

The best performance was obtained with the Rajski similarity metric clustered using spectral clustering or Louvain. Figures 16–18 illustrate the results. These and other figures in this section show the results when the matrices are pruned using k-nearest neighbors for $k = 30, 50, 100$, and 1000 (which stands in for no pruning). In this case, we see that the three-cluster configuration is relatively stable under pruning of the adjacency matrix.

Figure 19 shows that, for other, more divided, Congresses, our method produces the expected two clusters when the matrix is unpruned; however, the pruned matrix continues to produce three communities. We would like to identify which of these is the "best" clustering of the data, for example, by taking the clustering with the highest modularity. Unfortunately, this would be an apples-to-oranges comparison, because the value of the modularity depends on the adjacency matrix of the graph in addition to the clustering, and graphs with fewer edges will tend to have a higher modularity. We will revisit this question in our discussion of the German Bundestag data.

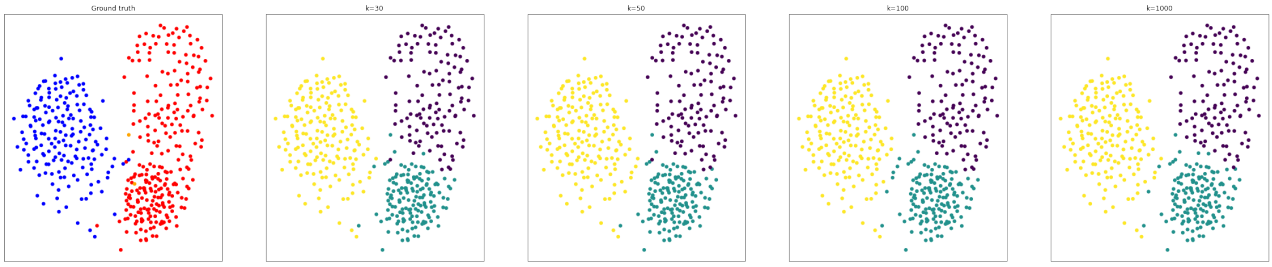


Figure 16: House of Representatives #81 (1949–1951), Louvain clustering with Rajski similarity matrix

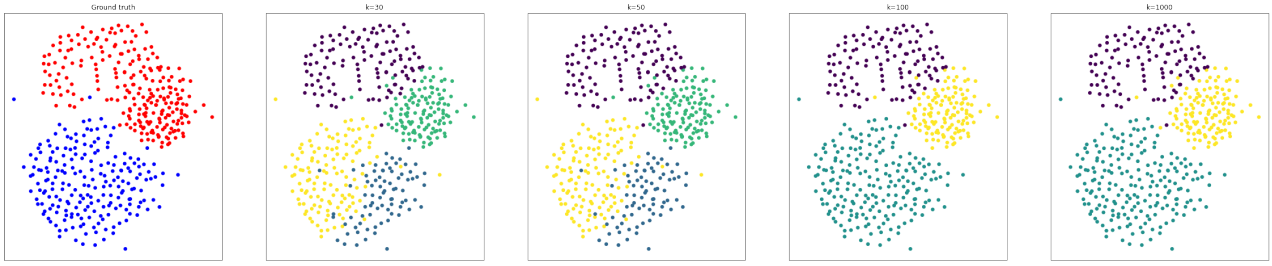


Figure 17: House of Representatives #85 (1957–1959), Louvain clustering with Rajski similarity matrix

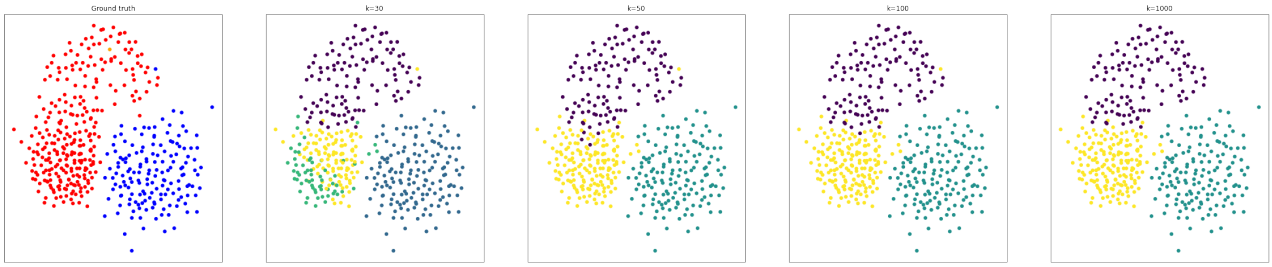


Figure 18: House of Representatives #86 (1959–1961), Louvain clustering with Rajski similarity matrix

Other distance metrics do not perform as well. Figure 20 is a representative sample of the results for other metrics. The clusters are unstable under changes in the value of k, and the third cluster includes both Democrats and Republicans, whereas we expect to see two factions of Republicans. Note also that tight clumping in the force-directed graphs suggests a weaker separation of the communities by other metrics.

Figure 21 shows that the Louvain clustering and spectral clustering, both with the Rajski similarity metric, perform similarly.

The United States has almost always had a two-party system; however, which two parties dominate changed several times in the nineteenth century. Those realignments—the collapse of the Federalist, Democratic-Republican, and Whig parties in turn—each appear in the findings of Waugh et al. as Congresses with medium-low modularity and three identifiable communities. Since after the Civil War, the modern party system of Democrat and Republican has been stable, however, the political orientations of these two parties have shifted. It is interesting that these intra-party realignments can be identified by the same network structural features as the inter-party realignments of the nineteenth century.

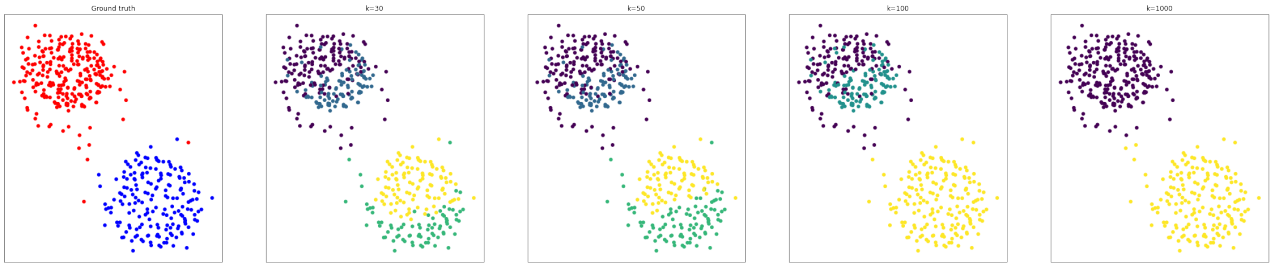


Figure 19: House of Representatives #111 (2009–2011), Louvain clustering with Rajski similarity matrix

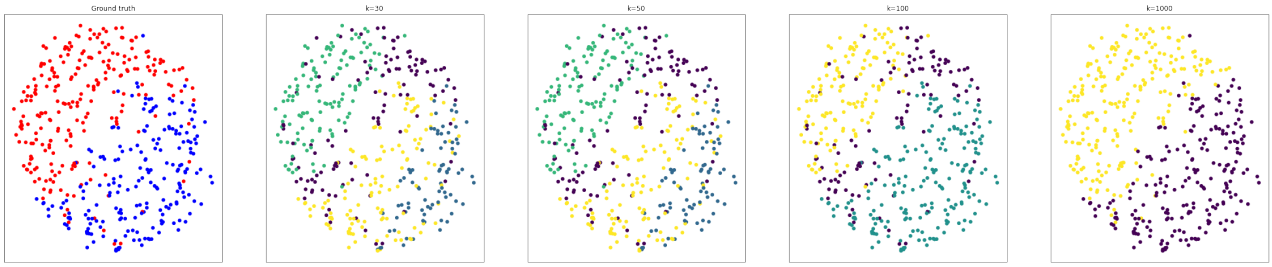


Figure 20: House of Representatives #85 (1957–1959), Louvain clustering with fractional similarity matrix

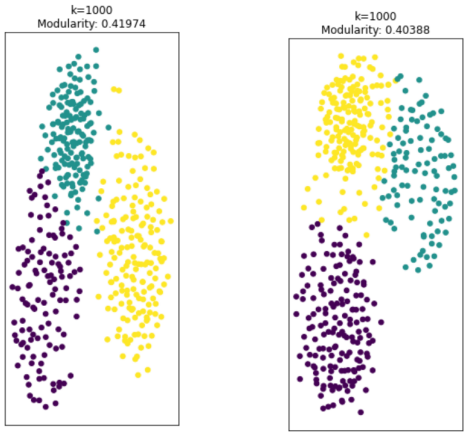


Figure 21: Congress #81 (1949-1951), Louvain (left) and spectral clustering (right) with Rajski similarity metric, unpruned. Note that these graphs were drawn with different seeds, hence the different geoemtries.

4.3 A multiclass look at the German Bundestag

The Bundestag is the German parliament established in 1949 by the constitution of the Federal Republic of Germany. The Prime Minister is elected by a majority of the members of the Bundestag. This typically requires a multiparty coalition. The Prime Minister then assembles a cabinet, which comprises the government in power. The Bundestag is elected every four years, unless sooner if there is a vote of no confidence in the Chancellor. This occurred twice, in 1982 (Bundestag 9) and 2005 (Bundestag 15). The current Bundestag is the nineteenth.

From Bundestag 4 to Bundestag 18, the following parties have held membership:

- Die Linke. Left wing. Historical descendent of the East German communist party.
- The Greens. Left wing.
- SPD (Social Democratic Party). Center left.
- FDP (Free Democratic Party). Liberal. Pro–free markets, privatization.
- CDU (Christian Democratic Union). Conservative.
- CSU (Christian Social Union). Conservative. Bavarian regional party.

The CSU is best thought of as a regional faction of the CDU, so I have combined the CDU and CSU for my clustering experiments.

In terms of the party composition of the Bundestag, we can identify three broad periods:

- The first three Bundestags were chaotic, and many tiny parties held membership.
- Bundestags 4–8 featured a mostly stable configuration of SPD vs. CU with FDP claiming a minority share.
- Since Bundestag 9, the party distribution in the Bundestag has been diverse. Two left wing parties have been represented and, recently, a right wing populist party AfD.

Our primary interest is in the modern period, whose party diversity makes it an interesting case study for multiclass clustering methods. We may consider the second period, when the Bundestag consisted of “merely" three parties, to be a control group.

As in our study of Congressional voting data, we assign to each legislator a vector of their roll call votes. We used roll call data from [26], which includes all roll call votes for Bundestag 1 through Bundestag 17. In the US Congress, almost all legislation is put to a roll call vote. But in the Bundestag and in most parliamentary systems, only a small fraction of the total legislation is put to a rollcall vote. Typically legislators vote anonymously. Those votes that are taken by name are unlikely to be a representative sample of all the votes of the Bundestag (see [16]). The Bundestags with the fewest recorded roll call votes, Bundestags 5 and 9, give strange and inconsistent results in our methods, suggesting that the records are either too small or too highly correlated. I’ve chosen to ignore those data in this report.

Tallies of the roll call votes recorded in each Bundestag are given below.

Bundestag	# roll call votes
1	133
2	169
3	46
4	37
5	24
6	38
7	55
8	59
9	26

Bundestag	# roll call votes
10	134
11	218
12	130
13	180
14	168
15	102
16	177
17	275

In the following experiments, the clusters identified by our algorithm are assigned parties in the Bundestag post-facto using a greedy assignment with payoff function

$$3 \times (\text{number of hits}) - (\text{number of misses}).$$

We use this assignment to measure the accuracy of the segmentation.

The Rajski and fraction metrics were found to most accurately parties in the Bundestag. Figure 22 shows the clusters obtained from these methods; compare figure 23, which shows the less accurate results of the cosine and SVD metrics.

Since the Rajski metric also gave good results with Congressional data, we will continue to use this metric to identify parties in the Bundestag. The choice of k when

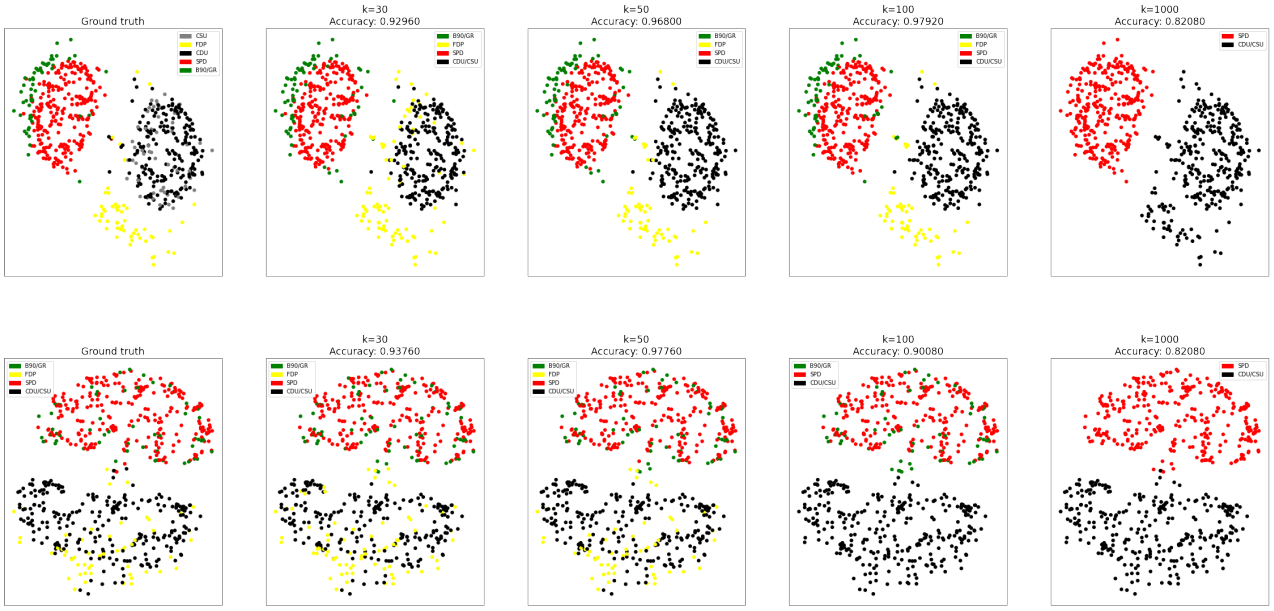


Figure 22: Bundestag #15 (2002-2005). SPD/Greens government. Louvain clustering with Rajski (top) and fractional (bottom) similarity matrices.

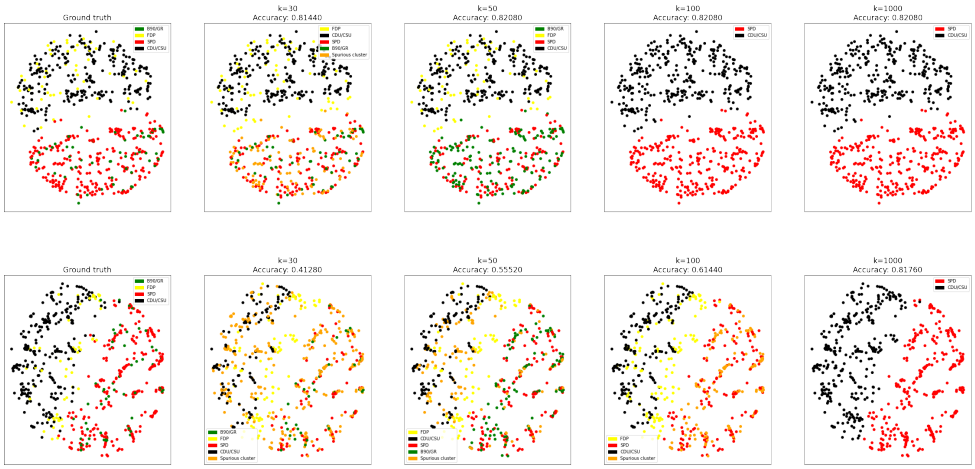
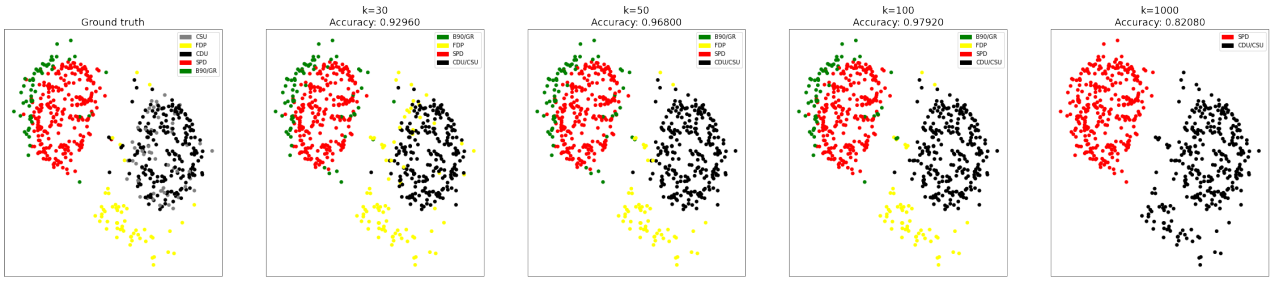
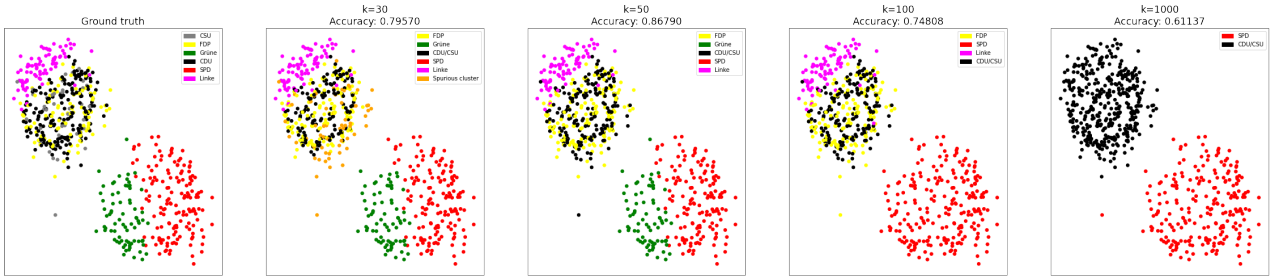


Figure 23: Bundestag #15 (2002-2005). SPD/Greens government. Louvain clustering with cosine (top) and SVD (bottom) similarity matrices.



(a) Bundestag #15 (2002-2005). SPD/Greens government.

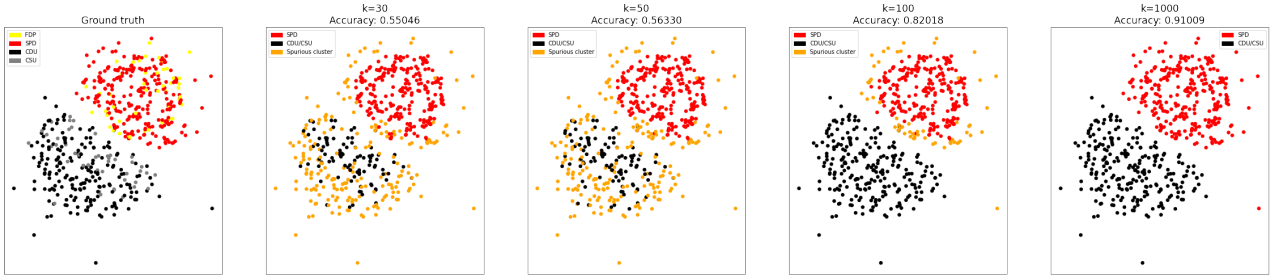


(b) Bundestag #17 (2009-2013). CU/SPD government.

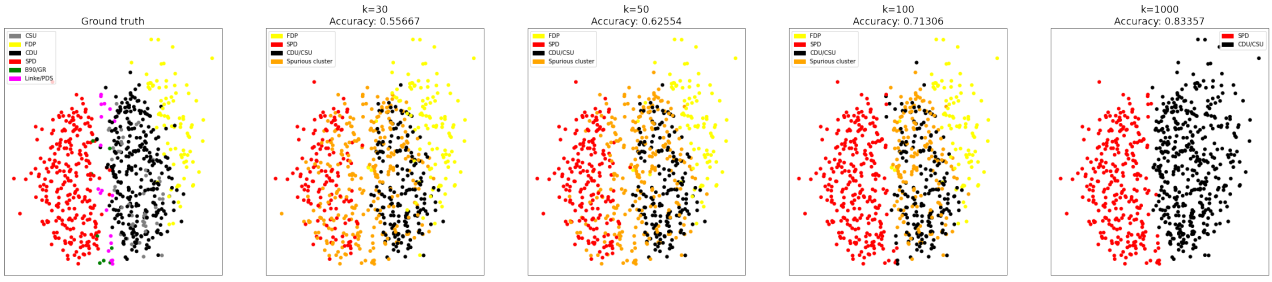
Figure 24: Two cases where $k = 50$ performs best. Louvain clustering with Rajski similarity matrix.

pruning the metric using k-means has a dramatic effect on the segmentation of our data. Figure 24 illustrates Bundestags for which an intermediate pruning parameter $k = 50$ gives the most accurate segmentation, while figure 25 illustrates Bundestags for which an unpruned similarity matrix gives the most accurate segmentation.

We would like our clustering algorithm to pick out the actual party configuration of the Bundestag. However, there are several obstacles to achieving perfect agreement with the empirical party distribution. One obstacle is that parties are under no obligation to organize themselves into maximum-modularity configurations. Parties organize around cultural, historical, and strategic concerns in addition to legislative priorities. For example, die Linke can trace its lineage back to the governing communist party of the German Democratic Republic; an MP who often votes with die Linke may nevertheless join the other left wing party, the Greens, if they find this legacy distasteful. (For this very reason, the SPD has historically refused to enter into a coalition with die Linke.) Figure 26 illustrates a case where modularity maximization does not give a good clustering of the MPs.



(a) Bundestag #7 (1972–1976). SPD/FDP government.



(b) Bundestag #12 (1990–1994). CU / FDP government.

Figure 25: Two cases where an unpruned adjacency matrix performs best. Louvain clustering with Rajski similarity matrix.

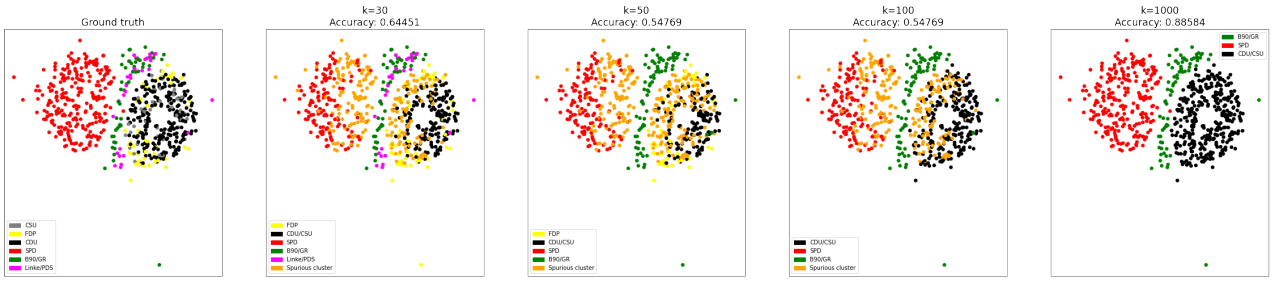


Figure 26: Bundestag #13 (1994–1998). CU/FDP government. Louvain clustering with Rajski similarity matrix.

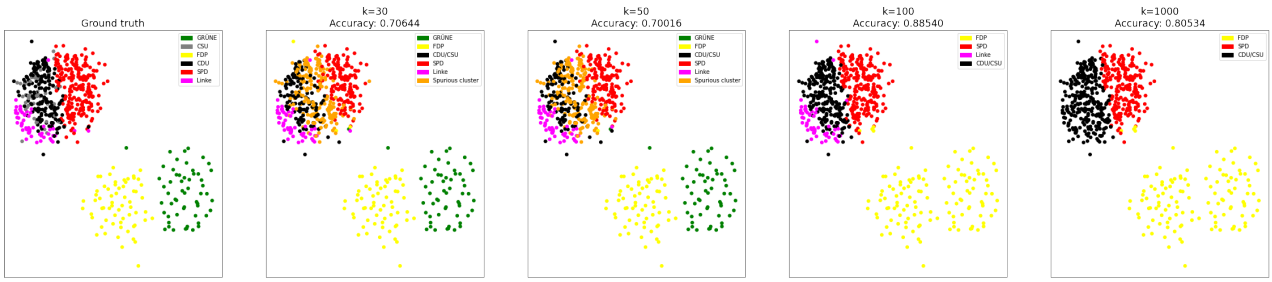


Figure 27: Bundestag #16 (2005–2009). CU/SPD government. Louvain clustering with Rajski similarity matrix.

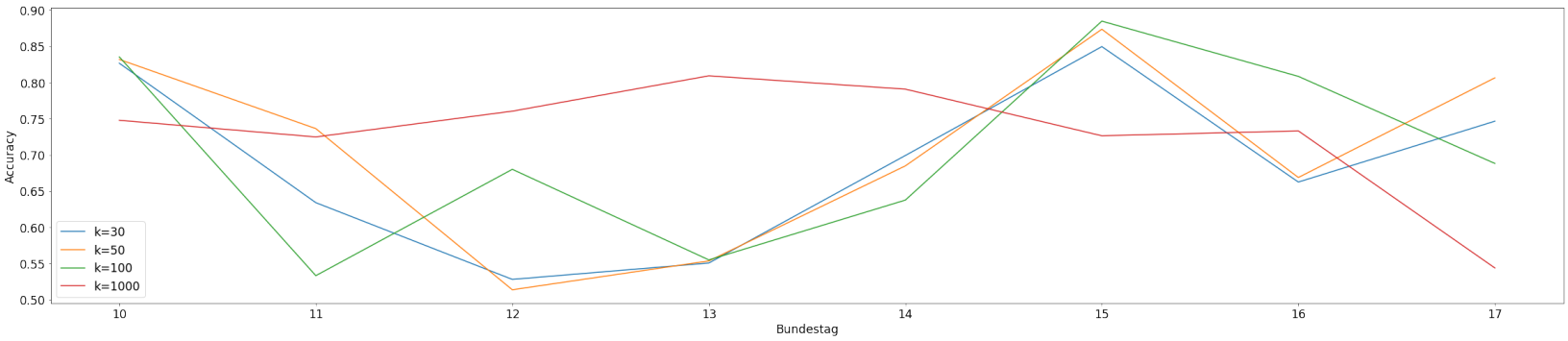


Figure 28: Performance of variously pruned matrices in our period of interest.

Another obstacle is that there are different salient scales on which the data can be clustered. At the largest scale, there is a binary classification problem: who is a member of the government and who is a member of the opposition? At the smallest scale, some sessions of the Bundestag feature minor parties with only a dozen or so representatives. The main parameter that controls scale in our experiments is the value of k with which we prune the similarity matrix using k-means. When we prune the similarity matrix, each node has degree $\leq k$. With no pruning, we frequently see just two clusters: government and opposition. Heavily pruned matrices are more sensitive to weak clusters. This makes it easier to pick out minor parties, but also produces spurious communities.

In the experiments above, we see that the optimal value of k varies from session to session of the Bundestag. We also see that in some sessions, such as in Figure 27, certain regions of the graph are best clustered with a small value of k, but other regions are best clustered by a large value. This suggests that the data is best clustered by assigning to each node a value of k corresponding to some natural local scale.

A crude way to accomplish this is to threshold the adjacency matrix. If we eliminate all edges with weight less than 0.5, then nodes with many close neighbors (which presumably belong to large parties) will retain a large degree, while nodes with fewer close neighbors will be more heavily pruned. Figure 29 shows that this represents a significant improvement over no pruning, and overall is our most accurate clustering method. However, it does not fully capture the benefits of $k = 50$ pruning in the cases where pruning gives good results.

In fact, thresholding is best understood as a supplement, rather than a substitute, for a judicious choice of pruning parameter k. Comparing figures 27 and 30 shows that with both thresholding and pruning (in this case, $k = 100$ works best), we obtain a very accurate segmentation that respects the different local scales of the data. However, the correct choice of k is not clear a priori, so a principled way of selecting the appropriate k remains to be found.

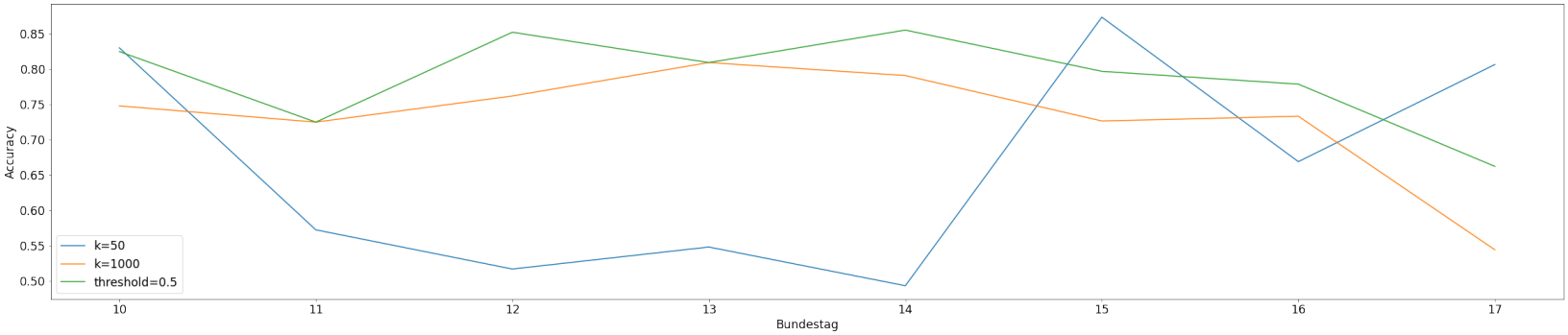


Figure 29: Accuracy of clustering with pruning, without pruning, and with thresholding.

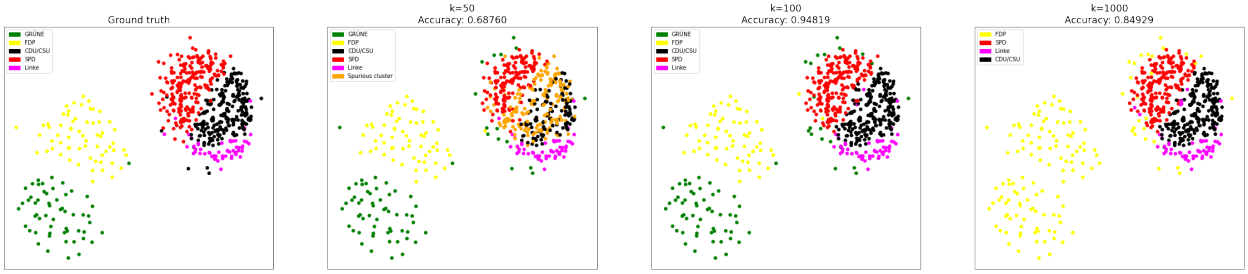


Figure 30: Bundestag #16 (2005–2009). CU/SPD government. Louvain clustering with thresholded Rajski similarity matrix. Case $k = 100$ is very accurate. Compare figure 27.

5 Kyung Ha: Legislative Council of Hong Kong and South Korean National Assembly

In this section, we focus on two non-US legislative systems, namely the legislative branch of Hong Kong and South Korea. As mentioned in section 2 we use the legislators voting records to generate different graphs and apply graph segmentation methods.

5.1 Legislative Council of Hong Kong

We use the voting data of the Legislative Council of Hong Kong from 2016 to 2021. Originally the Council was meant to last from October 1st, 2016 to September 30th, 2020 but was extended by the National People's Congress Standing Committee, which is the legislative body of the People's Republic of China. We collect the voting data from the Legislative Council of Hong Kong website [2].

5.1.1 2016 Hong Kong Election

Before we analyze the results we first need to understand how the Hong Kong election works and which parties are involved. All of the information presented in this section is based on the English Wiki of the 2016 Hong Kong legislative election page [1].

In total 70 legislators are elected. Half are elected directly through a proportional representation system and the other half is indirectly elected through interest-group-based functional constituencies (ex. Education, Tourism, Medical, Textiles, and Garment). Like in any other country there are candidates from multiple parties. In Hong Kong, there were 18 parties that got their candidates elected with an additional 12 independent legislators elected. These parties and independent legislators are classified as "Pro-Beijing camp", "Pan-Democrats", and "Localists" depending on their political stance. For example, the Pro-Beijing camp consists of parties Democratic Alliance for the Betterment and Progress of Hong Kong (DAB), Business and Professionals Alliance for Hong Kong (BPA), Hong Kong Federation of Trade Unions (FTU), etc. As the name suggests the Pro-Beijing camp is close to the policies of Mainland China, while the Pan-Democrats are more on the opposing side. The Localists is also against the policies from Beijing but more extreme than the Pan-Democrats, some advocating the independence of Hong Kong from Mainland China. The election results are given in Figure 31. In total

40 Pro-Beijing camp, 23 Pan-Democrats, 6 Localists and 1 non-aligned legislators have been elected.

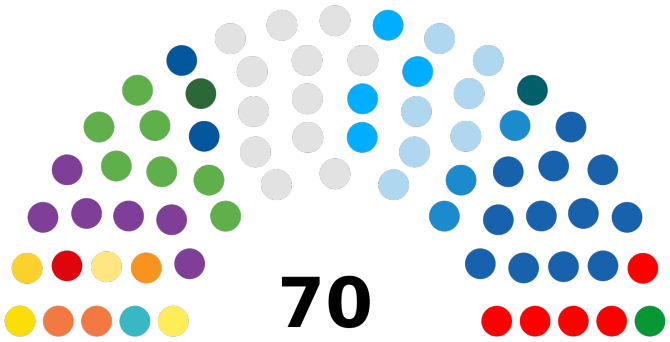


Figure 31: The 2016 Hong Kong election results. The red and blue colors on the right side are the Pro-Beijing camp parties. The left side are the Pan-Democrats and Localists parties (Localists are colored in yellow, light blue and red while the Pan-Democrats are colored in purple, green and dark blue. The grey seats in the middle are the independent (nonpartisan) legislators. . Figure from [1]

5.1.2 Numerical Results

We use the voting data from [2], generate various graphs using different metrics (Section 2), and use the GenLouvain algorithm (Section 4) to generate unsupervised classification results. We compare the results to the true partition of the legislators. As there are so many parties in the Hong Kong election, instead of associating each legislator with their party we associate them with the three camps: Pro-Beijing camp, Pan-Democrats and Localists. As the voting data from [2] does not mark each legislator's party or which camp they are associate with, we had to search each of their names and find their associated parties. This is manageable for legislators who were representing a specific party as we only need to know which camp their party is associated with. However, for legislators who were elected as independent, this is tricky as not much information about a specific Hong Kong legislator is available in English. Usually, there are only a few lines of information on the English Wiki and/or a news article with their name mentioned with other numerous legislators. Instead of trying to label all of the independent legislators we leave them in a pseudo-camp which we call "Independent". Remember that an Independent legislator are classified as one of the three groups according to their political opinion.

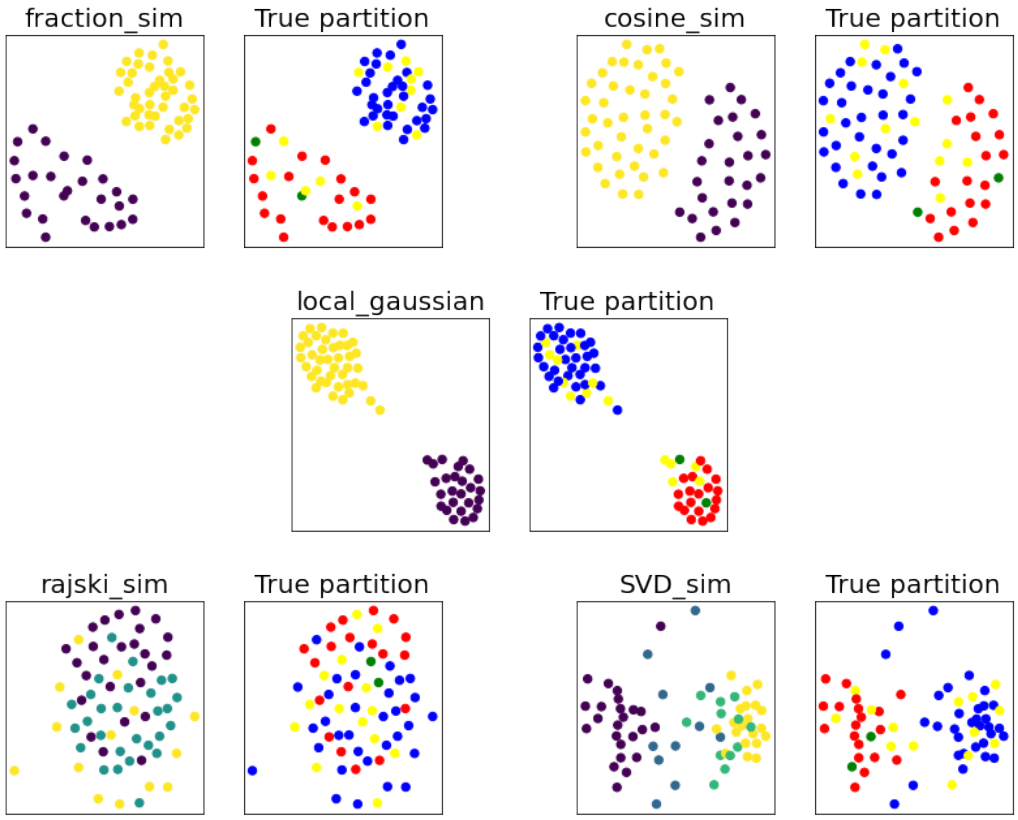


Figure 32: Graph partition results based on Hong Kong voting data. The partition results using different metrics to generate each graph. Each figure pair represents a different metric. For each pair, the left is an unsupervised partition given by the GenLouvain algorithm and the right is the true partition discussed in the text. For the true partition the colors for each camp is blue for Pro-Beijing , red for Pan-Democrat, green for Localist, and yellow for Independent.

The classification results are shown in figure 31. First focusing only on the partition results (left figure of each pair) we notice that the top three figures are well partitioned and the bottom two are not. For this reason, we only focus on the top three results. Comparing with the true partition (right figure of each pair) we observe that the top three indeed partition the legislators nicely into two groups, the Pro-Beijing in one group and the Pan-Democrat and Localist in the other group. Comparing the name of the legislators of these different partition results shows that the three different metrics actually yield the same partition results. This is an indication that the Legislative Council of Hong Kong in 2016 2021 is well divided.

Let us make a few remarks. First, as we exclude legislators who voted less than 25% of the time few legislators are not present on the graph. For example, two Localist legislators (Baggio Leung and Yau Wai-ching) were disqualified by the court on November 15th, 2016 almost right after the election, which is one reason why there are only two Localist present in the figures. Next, note that the Independent is scattered in both partitions. This is expected since some of the Independent legislators are actually on the Pro-Beijing side while others are on the Pan-Democrat and Localist side. The Independent legislators classified with the Pro-Beijing camp and the Independent legislators with the Pan-Democrat and Localist camps match with the number from [1]. This is another evidence that the algorithm has found the right partition.

5.1.3 Discussion

We have mentioned that as an English speaking user that it is hard to determine the political affiliation of an Independent legislator. However, using the results above we can determine which camp each legislator is associated with. For example we present some information of two Independent legislators posted on their English Wiki page [3, 4].

- Chan Kin-por : Represents the insurance functional constituency. Grew up in a poor family and didn't go to university, instead, he took a job in the insurance industry. Has restricted procedural delays by pan-democratic councilors
- Joseph Lee Kok-long : Represents the Health Services functional constituency. Received his Bachelor of Nursing degree from the La Trobe University, Australia in 1992. A member of the Pro-democracy camp.

From these alone we can make a good guess that Chan Kin-por is in the Pro-Beijing camp while Joseph Lee Kok-long is in the Pan-Democrat and Localist camp. Results in the previous section with the dots of the two legislators colored in orange and black confirm our guess (figure 33). Although for these two legislators we could have simply use the Wiki page to determine their camps some independent legislators do not have enough information. In this case, we can use our clustering method to determine which camp the independent legislator is associated with.

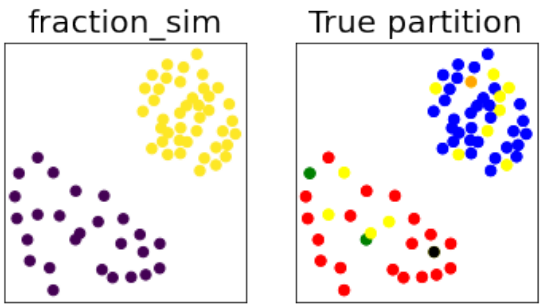


Figure 33: The first figure of figure 32 with Chan Kin-por colored in orange and Joseph Lee Kok-long colored in black on the right side.

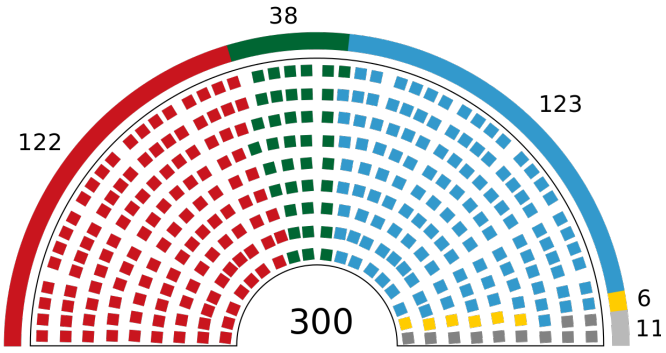


Figure 34: 2016 South Korea Legislator Election results

5.2 South Korean National Assembly

We use the voting data of the 20th South Korean National Assembly in 2016. The voting data was collected from the South Korean National Assembly website [10]. As our member who knows Korean did the research a lot of the sections are based on Korean websites. However most of the information can be found on the English Wiki, and Korean websites are only cited when necessary.

5.2.1 2016 South Korean Legislative Election

The 2016 South Korean Election which was held on April 13th, 2016 determined the members of the 20th South Korean National Assembly that started on May 30th, 2016 and ended on May 29th, 2020. In total 300 legislators were elected, 253 from first-past-the-post constituencies, and 47 from proportional party lists. Out of 26 parties, 4 parties had their candidates elected. These are starting from the right wing to the left wing Saenuri Party with 122 legislators, People's Party with 38 legislators, Democratic Party with 123 legislators, Justice Party with 6 legislators, and 11 Independent legislators (figure 34). All of the information above can be found on [5]

5.2.2 Numerical Results

Now we present the graph partitions results. Identical methods to obtain the Hong Kong graphs (figure 32) are applied to obtain figure 35. It is clear that none of the partitions obtained by the graph partition methods match the true partition according to the party. This is made clear if we group legislators according to their true parties with colors representing the partition results by our algorithm (figure 36). As discussed in the next section we think the problem is that there are not enough "no" votes in the data. One modification we tried is using the bills that only had at least one "no" vote in them. However, this modification did not change the results.

5.2.3 Discussion

Our natural question is why is the partition that the algorithm finds nothing like the true partition based on parties? Looking through the voting data of the legislators we discovered that there were not many "no" votes compared to "yes" votes. It turns out that in our voting data there are 81,533 "yes" votes while there are only 944 "no" votes [10]. For our algorithm, this means that each legislator is fairly similar to each other regardless of which party they belong to. Therefore the number and weights of the intra-party edges are similar to the inter-party edges. This turns out to be a general phenomenon for the South Korean National Assembly as during the term of the 20th National Assembly 8,799 bills were passed while only 2 failed to do so [6].

This may mean that South Korea has a unified legislative branch. However, looking through the [6] we find that this might be a naive conclusion. Out of the 24,141 bills

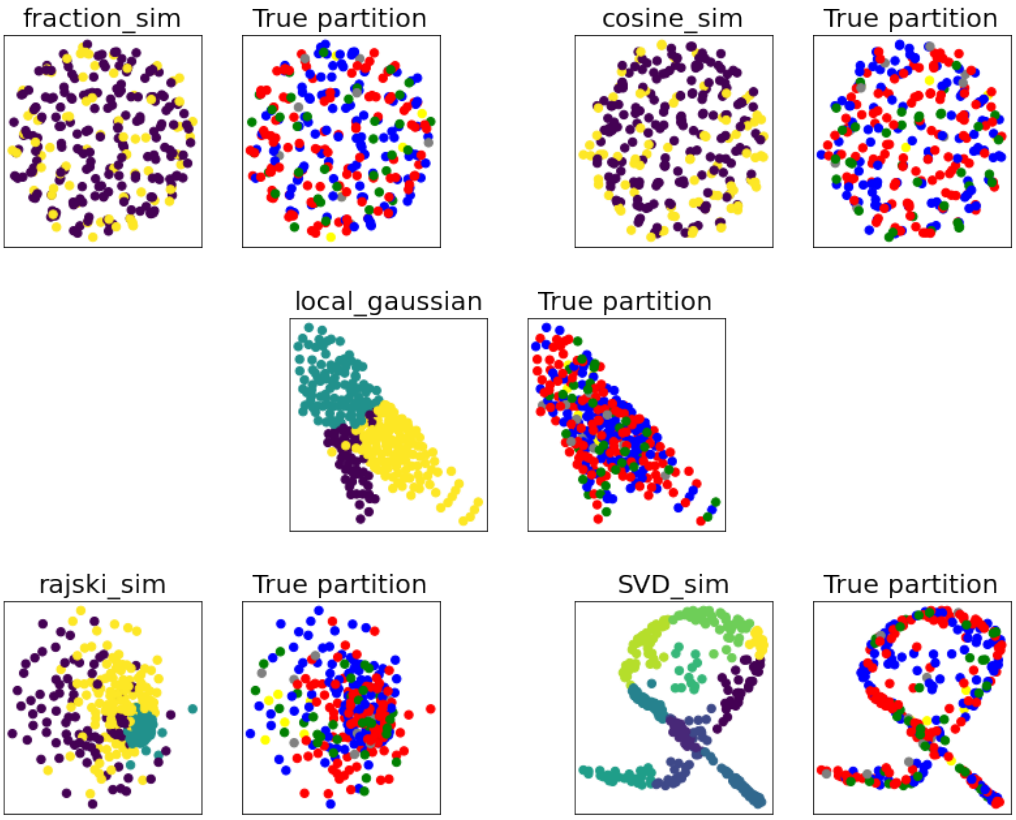


Figure 35: Graph partition results based on South Korea voting data. The partition results using different metrics to generate each graph. Each figure pair represents a different metric. For each pair, the left is an unsupervised partition given by the GenLouvain algorithm and the right is the true partition discussed in the text. For the true partition the colors for each party is red for Saenuri Party and blue for Democratic Party and green for People's Party and yellow for Justice Party. The Independent candidates are colored grey.

presented 15,340 bills have not even been able to vote. The largest contributor to this number is the 15,014 discarded bills due to term expiration. This means that over 60% of the effort of initiating a bill has been wasted. The main problem seems to be in the committees. For a bill to be passed in South Korea, the legislators have to initiate the bill and then the related committee decides whether to vote the bill during the regular session. If a bill is blocked by the committee (which is usually consisted of legislators from different parties) it is postponed. If a bill is not approved by the committee until the term of the National Assembly it is automatically discarded [11]. For such reason,

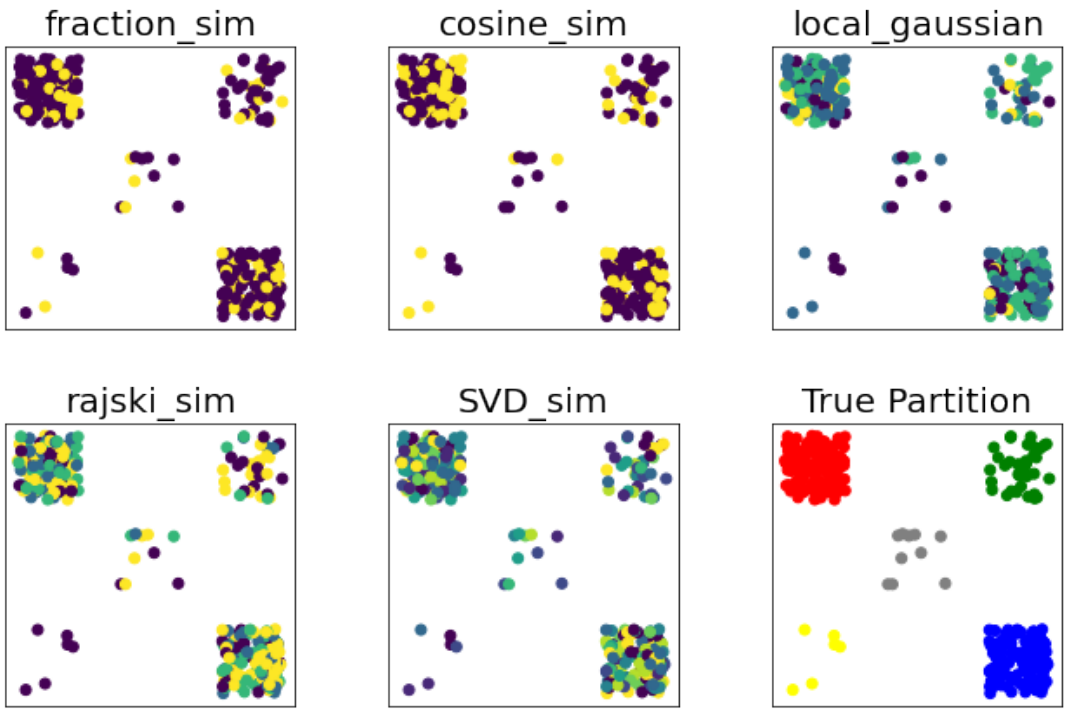


Figure 36: Graph partition results based on South Korea voting data grouped by parties. The true partition is given in the lower right figure.

we suspect that the bills that are actually voted during the sessions are the ones that have already passed their respective committees. This means that legislators from different parties have already agreed to pass them to the next step, indicating that there is a pre-voting phase that is not included in our voting record.

We present two reasons why we only use only the voting data of 2016 instead of the whole 20th National Assembly voting data. First is that downloading the data from the website is relatively difficult as it will not allow more than 10,000 voting records at a time (1 voting record amounts to the record of a vote of one legislator on one legislation) while there are 697,643 voting records in total. It is not simply downloading 70 csv files as once the search result has more than 10,000 records the downloading is impossible. One method that is viable in the future is searching the voting records by the name of the legislator which will require more than 300 searches and downloads [10].

The second is that a lot of parties have merged and divided after 2017. For example the Saenuri Party was divided into the Liberty Korea Party and Bareun Party in the start of 2017, while a year later in 2018 the People's Party and Bareun Party merged with Bareunmirae Party. And finally in 2020 the Liberty Korea Party and part of the Bareunmirae Party merged into the United Future Party. Upon all of this happening the legislators were changing their parties depending if they agreed or disagreed with the parties decision [8, 7, 9, 12]. It would be a lot of work to search how and when legislators moved parties and will likely give little insight into the results as the simulation results are not insightful for a different reason.

5.3 Conclusions

Using graph partition on Hong Kong and South Korea's voting data gave totally different results. In the case of Hong Kong, there was a clear cut indicating how much division is present in the Hong Kong Legislative Council. For South Korea, our algorithm fails. However, this also gave insight into South Korea's unique situation. In total, our algorithm is a useful tool that gives insights to the political system of different countries.

6 Thomas Tu: MBO Clustering

For my part of the project, I applied the MBO clustering scheme for modularity [17] to the voting data for the US Congress, the German Bundestag, and the South Korean and Hong Kong legislatures. I used a fidelity term and initial state derived using a basic spectral clustering scheme, and used volume preserving/auction MBO [19] to address issues with community sizes shrinking to zero.

6.1 MBO Background

The Ginzburg-Landau energy functional on graphs (with a semi-supervision fidelity term) was used in [13] for classification using a convex splitting scheme for gradient descent.

$$E(u) = \frac{\epsilon}{2} u \cdot Lu + \frac{1}{\epsilon} \int W(u) dx + F(u, u_0)$$

In [23], the double well potential $W(u)$ is replaced with a thresholding step, analogously to the original MBO scheme [24]. This MBO scheme on graphs is then used to perform semi-supervised clustering, following a similar semi-implicit Euler scheme.

6.2 Semi-Supervision

With the ground truth of the political affiliations being known, I could have sampled labels randomly from these to derive a partial fidelity, which could then be used for semi-supervision. However, here I instead decided to follow an approach from [22], where, lacking ground truth data for the hyperspectral plume video, the authors instead used a threshold on the eigenvectors of the graph Laplacian. The main reason I used this approach was because I had hoped to potentially see some division of the political parties into communities within the same party e.g. radicals vs. moderates. If I had imposed ground truth data on political affiliations, these groups would potentially be forced to end up in the same cluster. For the supervision, rather than using a threshold like in [22], I simply used the maximum value node for each eigenvector.

6.3 MBO Modularity

In [17], the authors extend the MBO scheme on graphs to optimize modularity, a network measure of community separation.

$$Q(g) = \frac{1}{2m}\sum_{i,j=1}^{N}(w_{ij} - \gamma\frac{k_i k_j}{2m})\delta(g_i, g_j)$$

Given a division of the members of a graph into communities, modularity measures the relative probability of an edge from one community member leading to another member of the same community, as proportional to the probability of that edge existing in a random null model. It lies in the range $[-\frac{1}{2}, 1]$, where 1 is perfectly separated (no edges between communities), 0 is perfectly random, and $-\frac{1}{2}$ indicates that connections are more likely to leave communities than stay within them. In [27], modularity score is used as a measure of party polarization: the more polarized the parties, the easier they are to segment into separate communities.

[17] reformulates modularity in terms of the Laplacian energy plus a balance term

$$Q(f) = 1 - \gamma - \frac{1}{2m}(|f|_{TV} - \gamma||f - mean(f)||_{l^2}^2)$$

$$|f|_{TV} = \frac{1}{2}\sum_{i,j=1}^{N} w_{ij}|f_i - f_j| = \frac{1}{2}\langle f, Lf\rangle$$

$$||f - mean(f)||_{l^2}^2 = \sum_{i=1}^{N} k_i \left| f_i - \frac{vol(A_i)}{2m}\right|^2, vol(A_i) = \sum_{i=1}^{N} k_i f_i$$

Maximizing modularity is thus equivalent to minimizing $|f|_{TV} - \gamma||f - mean(f)||_{l^2}^2$, where $|f|_{TV}$ is the Laplacian energy and $\gamma||f - mean(f)||_{l^2}^2$ is the balance term. This leads to a modified gradient descent equation (and corresponding numerical scheme.)

$$\frac{du}{dt} = -Lu - C_1\lambda(u - u_{fid}) + 2\gamma k^T(f - mean(f))$$

$$(I + dtL)u_{k+1} = u_k - dtC_1\lambda(u_k - u_{fid}) + 2\gamma dtk^T(f - mean(f))$$

6.4 Volume Preserving/Auction MBO

Traditional multiclass MBO thresholds by simply taking the max across communities, thresholding to corners of the simplex. However, in some cases, MBO can lead to some

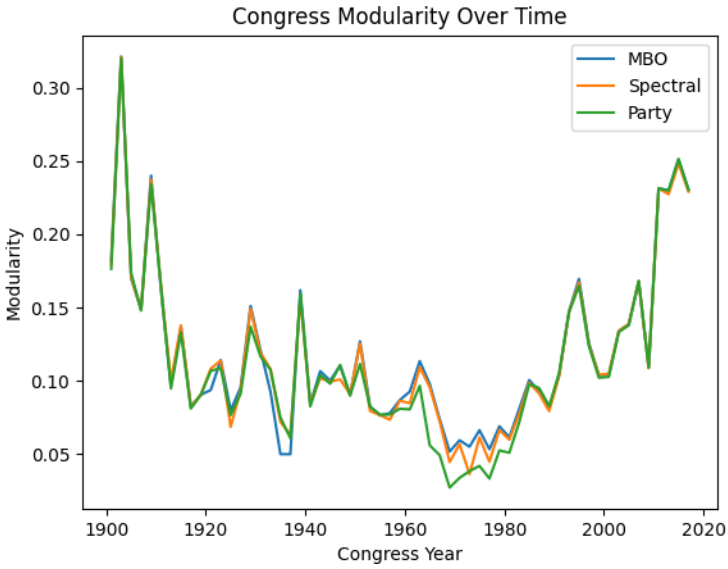


Figure 37: Plot of modularity over time for the US Congress using the fraction similarity matrix.

communities losing all their members. In [19], the authors use an auction scheme to enforce a minimum prescribed size for communities. This is done by auctioning members based on a "cost" equal to the value of their corresponding u. Here, I used a minimum community size equal to $\frac{N}{2\hat{n}}$, half of the size of a communities proportional share of the nodes.

6.5 Results

6.5.1 US Congress

I applied the MBO modularity scheme to the US Congress data, using the fraction and Rajski similarity matrices provided by Grace. In Section 6.5.1, I plot the modularity scores for each Congress over time, and in Section 6.5.1 I plot the optimal number of communities. The spectral clustering was found using a simple maximum over the eigenvectors corresponding to the smallest $\hat{n}$ eigenvalues of the graph Laplacian (not including the constant vector).

The optimal modularity runs mostly along party lines, except between 1960 and 1984. This agrees with the observations in [27]. In most cases (especially for recent

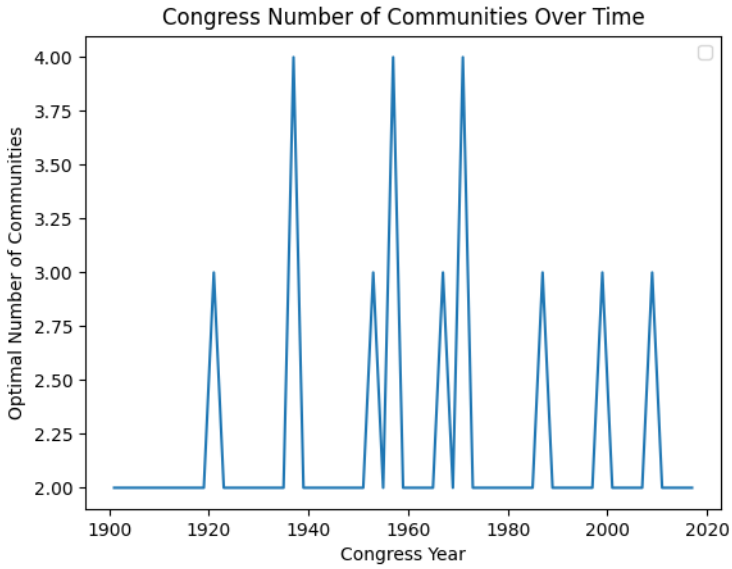


Figure 38: Plot of optimal number of communities for the US Congress over time, found using the fraction similarity matrix.

years), spectral and MBO return exactly the same two community partition, separating into one community of Republicans and another community containing Democrats and independents. These results are similar for the Rajski similarity matrices, which are plotted in Section 6.5.1 and Section 6.5.1; however, for these matrices, the simple spectral scheme often fails disastrously, leading to noisier results.

Initially, I attempted to impose a volume MBO condition to find three communities: two large communities, and one small community of independents. However, this approach failed; each time, the third community would instead select an essentially random set of nodes, with a very poor modularity score. I believe this to be due to modularity's resolution limit. Due to modularity relying on a null model of essentially random connections, it has difficulty finding communities of wildly differing sizes. Thus, it would be difficult or impossible to separate out any small community using modularity.

6.5.2 German Bundestag

I also applied MBO modularity to the Bundestag dataset using the matrices provided by Blaine. As with the US Congress data, both fraction and Rajski similarity matrices were used. The results are shown in Section 6.5.2, Section 6.5.2, Section 6.5.2, and Section 6.5.2.

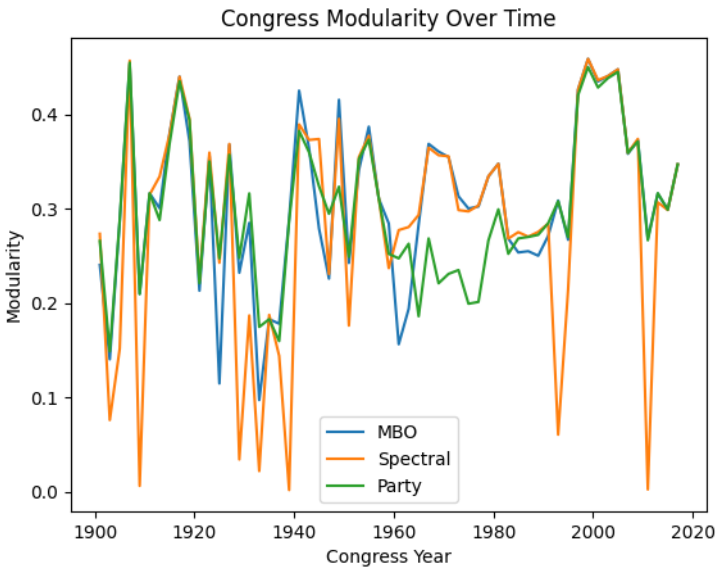


Figure 39: Plot of modularity over time for the US Congress using the Rajski similarity matrix.

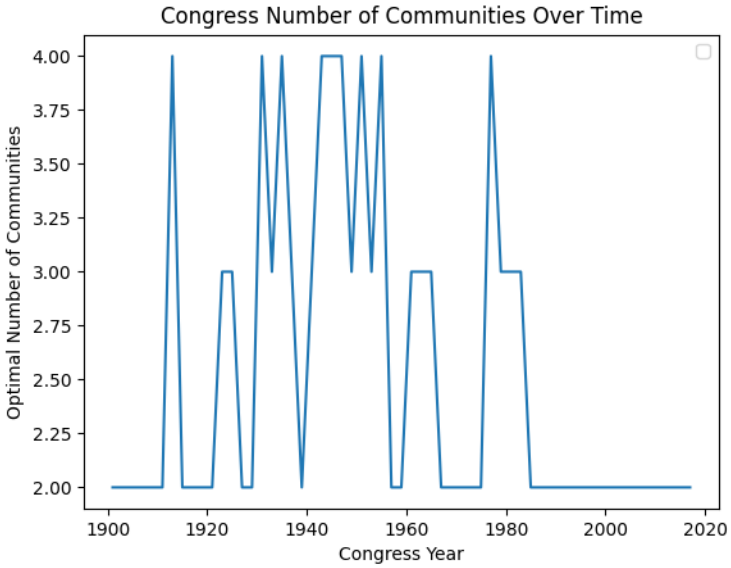


Figure 40: Plot of optimal number of communities for the US Congress over time, found using the Rajski similarity matrix.

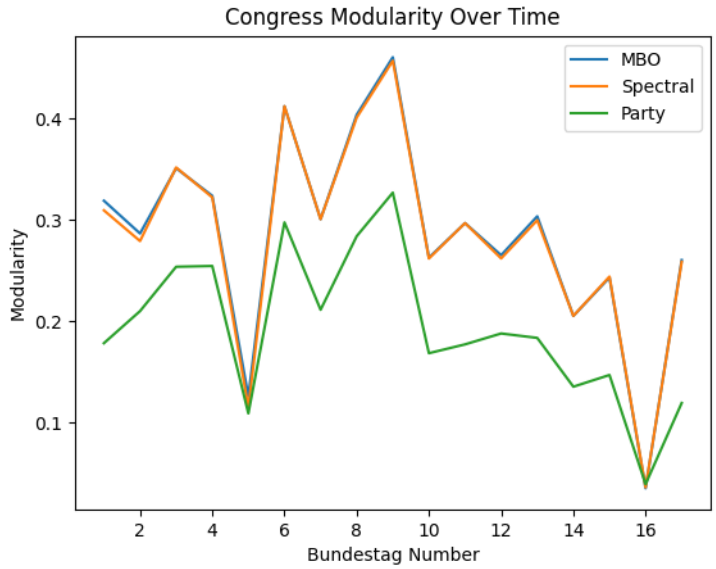


Figure 41: Plot of modularity over time for the German Bundestag using the fraction similarity matrix.

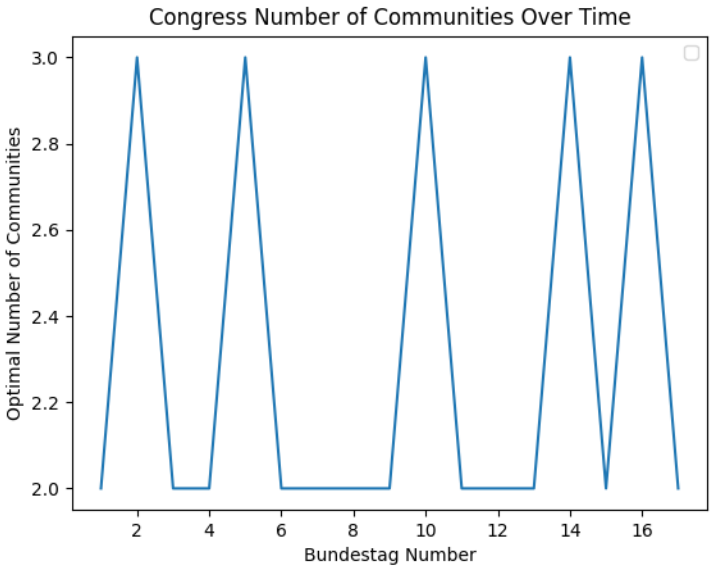


Figure 42: Plot of optimal number of communities for the German Bundestag over time, found using the fraction similarity matrix.

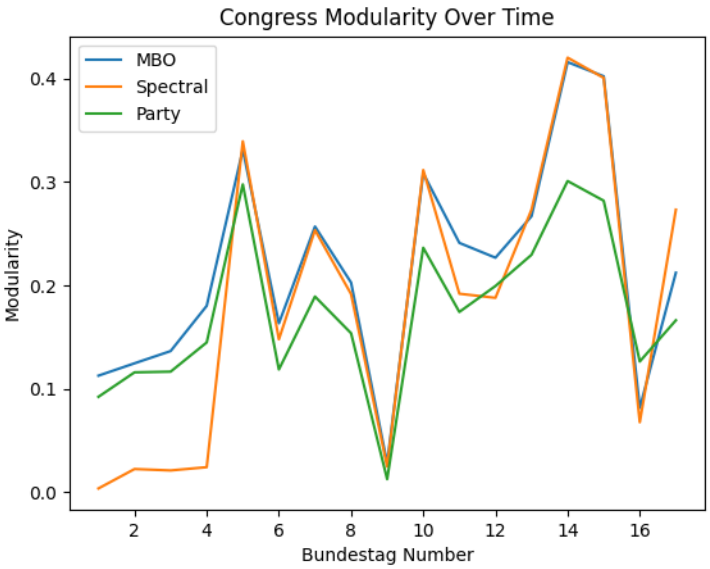


Figure 43: Plot of modularity over time for the German Bundestag using the Rajski similarity matrix.

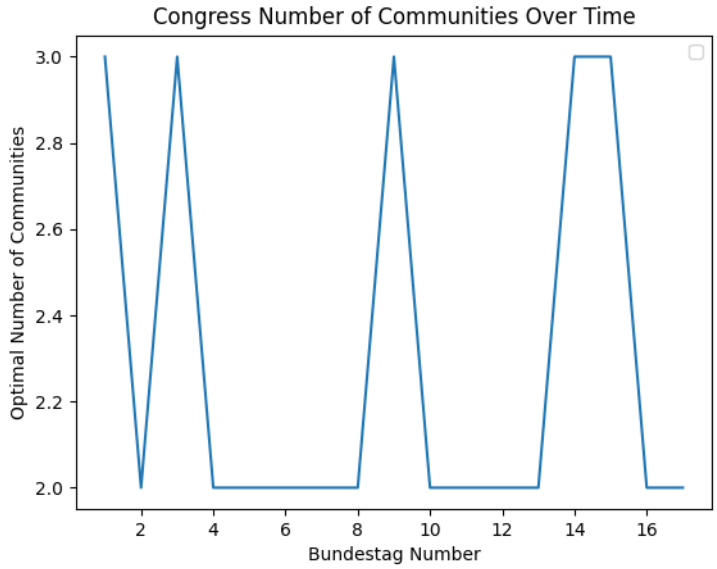


Figure 44: Plot of optimal number of communities for the German Bundestag over time, found using the Rajski similarity matrix.

Political Party	Pro-Beijing	Pan-Democrat	Localist	Independent
Community 1	32	0	0	9
Community 2	0	19	2	5

Table 1: Communities for the Hong Kong legislature found using MBO modularity, grouped by political party.

Clustering Method	Modularity Score
MBO	0.38744
Spectral	0.35842
Party	0.21083

Table 2: Modularity scores for each clustering approach.

The optimal division into communities divides the Bundestag not into political parties, but into coalitions. Thus, often the party modularity is much lower than the optimal modularity, as separate political parties are actually acting as one community.

6.5.3 Hong Kong Legislature

I also applied MBO modularity to the Hong Kong dataset using the similarity matrices provided by Kyung. Because the dataset only contains a single legislature, I did not plot the modularity over time for this dataset. Instead, I collected the data on which political parties each of the members of each community belonged to. These are listed in Table 1.

As with the Louvain approach, we see division into two communities representing the pro and anti-Beijing coalitions, with some independents split between the groups. In Table 2, I list the modularity scores for each clustering approach used. As with the German data, the party separation performs poorly due to the formation of coalitions.

6.5.4 South Korean Legislature

I also applied MBO modularity to the South Korean dataset using the similarity matrices provided by Kyung. Similarly to the Hong Kong data, this dataset also contains only a single legislature, and I instead show the data for the communities found in Table 3 and modularity scores in Table 4

Political Party Code	1	2	4	6	7	8
Community 1	2	0	1	0	5	2
Community 2	5	0	13	3	28	9
Community 3	3	1	15	3	20	8

Table 3: Communities for the South Korean legislature found using MBO modularity, grouped by political party

Clustering Method	Modularity Score
MBO	0.12949
Spectral	0.01370
Party	-0.01378

Table 4: Modularity scores for each clustering approach.

As with Kyung's approach, I found very poor clustering in the South Korean data, likely due to the same issues with the underlying vote data. Notably, the modularity score for the party separation is actually slightly negative, indicating that the community division according to parties performs worse than random. These low modularity scores give some indication of how difficult/impossible the data is to segment into well separated communities.

Bibliography

[1] 2016 Hong Kong legislative election Wiki. https://en.wikipedia.org/wiki/2016_Hong_Kong_legislative_election. Accessed: 2020-12-17.

[2] 6th Legislative Council of Hong Kong voting data. https://www.legco.gov.hk/general/english/open-legco/hc-202021.html. Accessed: 2020-12-17.

[3] Chan Kin-por (Hong Kong politician) Wiki page. https://en.wikipedia.org/wiki/Chan_Kin-por. Accessed: 2020-12-17.

[4] Joseph Lee (Hong Kong politician) Wiki page. https://en.wikipedia.org/wiki/Joseph_Lee_(Hong_Kong_politician). Accessed: 2020-12-17.

[5] South Korean 2016 legislator election Wiki. https://en.wikipedia.org/wiki/2016_South_Korean_legislative_election. Accessed: 2020-12-17.

[6] South Korean 20th National Assembly Bill Statistics. https://likms.assembly.go.kr/bill/stat/statFinishBillSearch.do. Accessed: 2020-12-17.

[7] South Korean Ahn Cheol-soo Wiki. https://en.wikipedia.org/wiki/Ahn_Cheol-soo#Bareunmirae_Party_%282018%E2%80%932019%29. Accessed: 2020-12-17.

[8] South Korean Bareun Party Wiki. https://en.wikipedia.org/wiki/Liberty_Korea_Party. Accessed: 2020-12-17.

[9] South Korean Liberty Korea Party Wiki. https://en.wikipedia.org/wiki/Liberty_Korea_Party. Accessed: 2020-12-17.

[10] South Korean National Assembly voting data of 2016. https://open.assembly.go.kr/portal/data/service/selectServicePage.do?infId=OPR1MQ000998LC12535&infSeq=1. Accessed: 2020-12-17.

[11] South Korean National Assembly Wiki in Korean. https://ko.wikipedia.org/wiki/%EB%8C%80%ED%95%9C%EB%AF%BC%EA%B5%AD_%EA%B5%AD%ED%9A%8C#%EC%9E%85%EB%B2%95%EC%97%90_%EA%B4%80%ED%95%9C_%EA%B6%8C%ED%95%9C. Accessed: 2020-12-17.

[12] South Korean United Future Party Wiki. https://en.wikipedia.org/wiki/People_Power_Party_(South_Korea)#Founding_congress. Accessed: 2020-12-17.

[13] Andrea Bertozzi and Arjuna Flenner. Diffuse interface models on graphs for classification of high dimensional data. *SIAM Review*, 58:293–328, 01 2016.

[14] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, oct 2008.

[15] Adam Boche, Jeffrey B. Lewis, Aaron Rudkin, and Luke Sonnet. The new voteview.com: preserving and continuing keith poole's infrastructure for scholars, students and observers of congress. *Public Choice*, 176(1):17–32, Jul 2018.

[16] Clifford J. Carrubba, Matthew Gabel, Lacey Murrah, Ryan Clough, Elizabeth Montgomery, and Rebecca Schambach. Off the record: Unrecorded legislative votes, selection bias and roll-call vote analysis. *British Journal of Political Science*, 36(4):691–704, 2006.

[17] Huiyi Hu, Thomas Laurent, Mason A Porter, and Andrea L Bertozzi. A method based on total variation for network modularity optimization using the mbo scheme. *SIAM Journal on Applied Mathematics*, 73(6):2224–2246, 2013.

[18] Huiyi Hu, Justin Sunu, and Andrea L. Bertozzi. Multi-class graph mumford-shah model for plume detection using the mbo scheme. In Xue-Cheng Tai, Egil Bae, Tony F. Chan, and Marius Lysaker, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 209–222, Cham, 2015. Springer International Publishing.

[19] Matt Jacobs, Ekaterina Merkurjev, and Selim Esedoḡlu. Auction dynamics: A volume constrained mbo scheme. *Journal of Computational Physics*, 354:288–310, 2018.

[20] Aleks Jakulin, Wray Buntine, Timothy M. La Pira, and Holly Brasher. Analyzing the u.s. senate in 2003: Similarities, clusters, and blocs. *Political Analysis*, 17(3):291–310, 2009.

[21] Jeffrey B. Lewis, Keith Poole, Howard Rosenthal, Adam Boche, Aaron Rudkin, and Luke Sonnet. Voteview: Congressional roll-call votes database., 2020.

[22] Zhaoyi Meng, Ekaterina Merkurjev, Alice Koniges, and Andrea L Bertozzi. Hyperspectral image classification using graph clustering methods. *Image Processing On Line*, 7:218–245, 2017.

[23] Ekaterina Merkurjev, Tijana Kostic, and Andrea L Bertozzi. An mbo scheme on graphs for classification and image processing. *SIAM Journal on Imaging Sciences*, 6(4):1903–1930, 2013.

[24] Barry Merriman, James Kenyard Bence, and Stanley Osher. *Diffusion generated motion by mean curvature*. Department of Mathematics, University of California, Los Angeles, 1992.

[25] Mason A. Porter, Peter J. Mucha, M. E. J. Newman, and Casey M. Warmbrand. A network analysis of committees in the u.s. house of representatives. *Proceedings of the National Academy of Sciences*, 102(20):7057–7062, 2005.

[26] Ulrich Sieberer, Thomas Saalfeld, Tamaki Ohmura, Henning Bergmann, and Stefanie Bailer. Roll-call votes in the german bundestag: A new dataset, 1949–2013. *British Journal of Political Science*, 50(3):1137–1145, December 2018.

[27] Andrew Scott Waugh, Liuyi Pei, James H. Fowler, Peter J. Mucha, and Mason A. Porter. Party polarization in congress: A network science approach. 2011.

[28] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, NIPS'04, page 1601–1608, Cambridge, MA, USA, 2004. MIT Press.