# Plural Decision-Making on Graphs

Joel Miller

December 2024

## 1 Introduction

In this document, we explore techniques for using social graphs as input to plural public choice mechanisms — mechanisms that attempt to find public choices favored by individuals from broad social strata (i.e. that prioritize "agreement across difference"). These experiments will use trust graph used by Circles [1]

Many different community detection algorithms exist (for a survey see [2]). For a given graph, different community detection algorithms will divide the graph in different ways, thus yielding different notions of what counts as "difference".

However, measuring the performance of various algorithms is tricky because said measurement must always be done relative to a baseline, and in this setting, choosing a baseline means taking a normative stance on what a communities "actually" look like. One's normative assumptions about what constitutes a community will naturally skew the measurement of the best-performing community detection algorithm. For example, let's say I consider an algorithm "more plural" if it prioritizes viewpoints from diverse $k$-cliques. Then I should not be surprised when a community detection algorithm that calculates $k$-cliques performs the best in my tests.

Since I do not want to take a normative stance on what "actually" constitutes a community, I will instead shift my focus to an examination of how different community detection algorithms overlap and interact.

In other words, each community detection algorithm defines both a *hypothesis* about what communities exist in a graph, and a *measurement* of those communities. My interest in this document is to understand how well the measurements of one community detection algorithm sync up with the hypothesis of the another.

Put another way, suppose we have a community detection algorithm $X$. Suppose the members of the graph participate in a vote. We can simulate a voting behavior by having agents vote according to which community they belong to (as defined by $X$), possibly with some noise. Now, using the communities from $X$ as input for a plural decision making algorithm yields some decision – we can call this the "baseline decision". Then, for a different community detection algorithm $Y$, we can ask if a plural decision making algorithm using the communities defined by $Y$ yields the same decision as our baseline. By repeating this experiment many times with different voting patterns, for different pairs of $X$ and $Y$, we can see which (if any) community detection algorithms give pural results when ran on voting data generated under other hypotheses about what constitutes a community.

## 2 Technical Setup

The plural decision making algorithm we will use is **Cluster Match**, a variant of quadratic funding. In Cluster Match, participants make contributions in favor of various proposals, and returns a "score" for that proposal. Formally cluster Match takes the following as input:

- $N$, a set of agents.
- $M$, a set of groups.

[1] `joincircles.net`

- A vector $c$ of non-negative contributions.
- A $|M| \times |N|$ matrix $W$ describing the weight of each agent in each group, normalized so that each agent's weights across all groups sum to 1.
- vector $w$ describing the weight of each group.

Then cluster match calculates the score for that proposal as

$$\left(\sum_{g \in G} w_g \sqrt{W_g \cdot c}\right)^2$$

where $W_g$ is the row of $W$ corresponding to group $g$ and $\cdot$ is the dot product operation.

The scores from cluster match can be used in a variety of ways. In our experiments, we will opt for a simplified setup where, given multiple proposals, we use cluster match to calculate a score for each proposal, and then declare the proposal with the highest score as the "winner".

## 2.1 Community Detection Methods

Here, we list the community detection methods we study. All but the last method give partitions of the graph where each individual is in exactly one group. For all of these methods except spectral polysection, set $w = \mathbf{1}$.

**No clustering (NC).** Each individual is assigned to their own cluster.

**Greedy modularity maximization (GMM).** Described in [1].

**Label propagation (LP).** A randomized algorithm described in [3].

**Repeated spectral bisection (RSB).** Recursively bisect the graph using its Feidler vector, stopping when either the recursion depth limit is reached or a sub-graph has 10 or fewer members. I used recursion depth limits of 5, 6, and 7, yielding partitions with 32, 60, and 108 communities respectively.

**Spectral Polysection (SP).** Unlike the other methods, this one uses $w$ and $W$ more carefully. Take the 2nd through $k+1$th eigenvectors and eigenvalues of the graph's laplacian matrix, in increasing order of the eigenvalues. Let $\lambda$ denote the vector of eigenvalues. Then the $i$th eigenvector $v_i$ describes two groups, $i_{\mathsf{pos}}$ and $i_{\mathsf{neg}}$, with weights of $w_{i_{\mathsf{pos}}} = w_{i_{\mathsf{neg}}} = 1/\lambda_i$. Furthermore, for an agent $j$, define $W_{i_{\mathsf{pos}},j}$ as $v_{i,j}$ if $v_{i,j} > 0$ and 0 otherwise. Similarly, define $W_{i_{\mathsf{pos}},j}$ as $-v_{i,j}$ if $v_{i,j} < 0$ and 0 otherwise. In other words, each eigenvector creates two groups: agents whose entries in the vector have positive values, and agents in whose entries in the vector have negative values. Each agent is in one of those groups, with a weight corresponding to absolute value of their entry in the vector. Finally, the groups themselves are weighted according to the reciprocal of the corresponding eigenvalue, so that the smallest-valued eigenvectors (i.e. the Feidler vector) have more weight in the decision-making process. As a final post-processing step, normalize $W$.

We tested SP with different values of $k$–5, 10, 25, 50, 100, and lastly using every eigenvector with an eigenvalue of at most 1 (of which there were 1,634).

## 2.2 Graph Preparation

We obtained the circles trust graph for use in these experiments. Circles is a community-based UBI platform which allows users to create trust links between each other. These trust links are uni-directional, i.e., user $A$ may trust user $B$ without $B$ necessarily trusting $A$. To create the graph used in our experiments, we started with the full graph of circles trust attestations. Then, we created an undirected graph by removing all edges

| | NC | GM | LP | RSB-5 | RSB-6 | RSB-7 | SP-5 | SP-10 | SP-25 | SP-50 | SP-100 | SP-1634 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NC | 1.0 | 0.748 | 0.76 | 0.78 | 0.736 | 0.718 | 0.52 | 0.52 | 0.52 | 0.52 | 0.52 | 0.52 |
| GM | 0.894 | 1.0 | 0.828 | 0.826 | 0.802 | 0.786 | 0.488 | 0.488 | 0.488 | 0.488 | 0.488 | 0.488 |
| LP | 0.8 | 0.832 | 1.0 | 0.83 | 0.82 | 0.804 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |
| RSB-5 | 0.87 | 0.904 | 0.848 | 1.0 | 0.98 | 0.942 | 0.532 | 0.532 | 0.532 | 0.532 | 0.532 | 0.532 |
| RSB-6 | 0.854 | 0.88 | 0.812 | 0.972 | 1.0 | 0.964 | 0.486 | 0.486 | 0.486 | 0.486 | 0.486 | 0.486 |
| RSB-7 | 0.834 | 0.85 | 0.792 | 0.932 | 0.968 | 1.0 | 0.52 | 0.52 | 0.52 | 0.52 | 0.52 | 0.52 |
| SP-5 | 0.506 | 0.504 | 0.494 | 0.502 | 0.506 | 0.506 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-10 | 0.468 | 0.484 | 0.484 | 0.474 | 0.478 | 0.484 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-25 | 0.526 | 0.528 | 0.534 | 0.536 | 0.53 | 0.52 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-50 | 0.47 | 0.458 | 0.458 | 0.476 | 0.486 | 0.484 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-100 | 0.524 | 0.508 | 0.51 | 0.518 | 0.52 | 0.528 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-1634 | 0.48 | 0.478 | 0.464 | 0.496 | 0.496 | 0.498 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |

Table 1: Cross-algorithm agreement with seeding according to an even vote split.

| | NC | GM | LP | RSB-5 | RSB-6 | RSB-7 | SP-5 | SP-10 | SP-25 | SP-50 | SP-100 | SP-1634 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NC | 1.0 | 0.668 | 1.0 | 0.586 | 0.62 | 0.746 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| GM | 0.856 | 1.0 | 0.82 | 0.828 | 0.826 | 0.82 | 0.778 | 0.778 | 0.778 | 0.778 | 0.778 | 0.778 |
| LP | 0.866 | 0.782 | 1.0 | 0.69 | 0.764 | 0.86 | 0.998 | 0.998 | 0.998 | 0.998 | 0.998 | 0.998 |
| RSB-5 | 0.882 | 0.864 | 0.842 | 1.0 | 0.966 | 0.948 | 0.772 | 0.772 | 0.772 | 0.772 | 0.772 | 0.772 |
| RSB-6 | 0.86 | 0.84 | 0.812 | 0.944 | 1.0 | 0.978 | 0.84 | 0.84 | 0.84 | 0.84 | 0.84 | 0.84 |
| RSB-7 | 0.814 | 0.798 | 0.806 | 0.846 | 0.928 | 1.0 | 0.894 | 0.894 | 0.894 | 0.894 | 0.894 | 0.894 |
| SP-5 | 0.692 | 0.7 | 0.698 | 0.752 | 0.752 | 0.75 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-10 | 0.704 | 0.748 | 0.744 | 0.748 | 0.748 | 0.746 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-25 | 0.866 | 0.888 | 0.846 | 0.83 | 0.832 | 0.826 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-50 | 0.972 | 0.982 | 0.98 | 0.95 | 0.954 | 0.952 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-100 | 0.996 | 0.998 | 1.0 | 0.994 | 0.996 | 0.996 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-1634 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |

Table 2: Cross-algorithm agreement with seeding according to an 33%-66% vote split.

# 3 Experiments

For each pair of community detection algorithms $(X, Y)$ and a variety of noise levels, we ran 100 trials where we seeded the graph with votes corresponding to the communities as defined by $X$, attained the result of cluster match using the communities defined by $X$, and then checked if cluster match gave the same output when using the communities as defined by $Y$. The number of agreements are listed below, where the column denotes algorithm $X$ and the row denotes algorithm $Y$.

All voting was simulated using just two options for simplicity. In the vote seeding process, we picked a random preferred option for each community, and had all its members vote for that option with a contribution strength of 1. Then, depending on the noise level (a real number less than one), we flipped each individual's vote with that probability. We used noise levels of 0, 0.25, and 0.5.

Table 3: Cross-algorithm agreement with seeding according to an 80%-20% vote split.

| | NC | GM | LP | RSB-5 | RSB-6 | RSB-7 | SP-5 | SP-10 | SP-25 | SP-50 | SP-100 | SP-1634 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NC | 1.0 | 0.668 | 1.0 | 0.586 | 0.62 | 0.746 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| GM | 0.856 | 1.0 | 0.82 | 0.828 | 0.826 | 0.82 | 0.778 | 0.778 | 0.778 | 0.778 | 0.778 | 0.778 |
| LP | 0.866 | 0.782 | 1.0 | 0.69 | 0.764 | 0.86 | 0.998 | 0.998 | 0.998 | 0.998 | 0.998 | 0.998 |
| RSB-5 | 0.882 | 0.864 | 0.842 | 1.0 | 0.966 | 0.948 | 0.772 | 0.772 | 0.772 | 0.772 | 0.772 | 0.772 |
| RSB-6 | 0.86 | 0.84 | 0.812 | 0.944 | 1.0 | 0.978 | 0.84 | 0.84 | 0.84 | 0.84 | 0.84 | 0.84 |
| RSB-7 | 0.814 | 0.798 | 0.806 | 0.846 | 0.928 | 1.0 | 0.894 | 0.894 | 0.894 | 0.894 | 0.894 | 0.894 |
| SP-5 | 0.692 | 0.7 | 0.698 | 0.752 | 0.752 | 0.75 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-10 | 0.704 | 0.748 | 0.744 | 0.748 | 0.748 | 0.746 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-25 | 0.866 | 0.888 | 0.846 | 0.83 | 0.832 | 0.826 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-50 | 0.972 | 0.982 | 0.98 | 0.95 | 0.954 | 0.952 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-100 | 0.996 | 0.998 | 1.0 | 0.994 | 0.996 | 0.996 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-1634 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |

Table 4: Cross-algorithm agreement with seeding according to an 50%-50% vote split, and 25% noise.

# References


[1] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. "Finding community structure in very large networks". In: *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 70.6 (2004), p. 066111.

[2] Jiakang Li et al. "A comprehensive review of community detection in graphs". In: *Neurocomputing* 600 (2024), p. 128169. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2024.128169. URL: https://www.sciencedirect.com/science/article/pii/S0925231224009408.

[3] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. "Near linear time algorithm to detect community structures in large-scale networks". In: *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 76.3 (2007), p. 036106.


# A Effects of noise

We also ran the experiments described in section 3 with added noise. In these experiments, votes are first seeded according to the communities individuals reside in, but then with some chance each individual flips their vote. We tested out "noise levels" of 12.5% and 25%, i.e., we ran simulations where each individual had a 12.5% or 25% chance of flipping their vote, for each of the voting patterns described above (50%-50%, 33%-66%, and 20%-80%). Results are listed in tables 4, 5, 6, 7, 8 and 9.

| | NC | GM | LP | RSB-5 | RSB-6 | RSB-7 | SP-5 | SP-10 | SP-25 | SP-50 | SP-100 | SP-1634 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NC | 1.0 | 0.002 | 0.062 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| GM | 0.826 | 1.0 | 0.826 | 0.866 | 0.854 | 0.832 | 0.478 | 0.478 | 0.478 | 0.478 | 0.478 | 0.478 |
| LP | 0.82 | 0.468 | 1.0 | 0.458 | 0.462 | 0.512 | 0.74 | 0.74 | 0.74 | 0.74 | 0.74 | 0.74 |
| RSB-5 | 0.752 | 0.848 | 0.776 | 1.0 | 0.966 | 0.926 | 0.438 | 0.438 | 0.438 | 0.438 | 0.438 | 0.438 |
| RSB-6 | 0.75 | 0.874 | 0.8 | 0.924 | 1.0 | 0.972 | 0.422 | 0.422 | 0.422 | 0.422 | 0.422 | 0.422 |
| RSB-7 | 0.77 | 0.848 | 0.788 | 0.896 | 0.926 | 1.0 | 0.478 | 0.478 | 0.478 | 0.478 | 0.478 | 0.478 |
| SP-5 | 0.644 | 0.502 | 0.552 | 0.49 | 0.504 | 0.51 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-10 | 0.724 | 0.614 | 0.65 | 0.568 | 0.574 | 0.582 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-25 | 0.862 | 0.784 | 0.784 | 0.648 | 0.654 | 0.654 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-50 | 0.962 | 0.93 | 0.952 | 0.89 | 0.892 | 0.896 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-100 | 0.99 | 0.992 | 1.0 | 0.982 | 0.984 | 0.982 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-1634 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |

Table 5: Cross-algorithm agreement with seeding according to an 33%-66% vote split, and 25% noise.

Table 6: Cross-algorithm agreement with seeding according to an 80%-20% vote split, and 25% noise.

| | NC | GM | LP | RSB-5 | RSB-6 | RSB-7 | SP-5 | SP-10 | SP-25 | SP-50 | SP-100 | SP-1634 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NC | 1.0 | 0.678 | 0.74 | 0.776 | 0.746 | 0.756 | 0.498 | 0.498 | 0.498 | 0.498 | 0.498 | 0.498 |
| GM | 0.836 | 1.0 | 0.788 | 0.828 | 0.808 | 0.79 | 0.522 | 0.522 | 0.522 | 0.522 | 0.522 | 0.522 |
| LP | 0.742 | 0.75 | 1.0 | 0.734 | 0.726 | 0.73 | 0.512 | 0.512 | 0.512 | 0.512 | 0.512 | 0.512 |
| RSB-5 | 0.864 | 0.832 | 0.81 | 1.0 | 0.966 | 0.93 | 0.512 | 0.512 | 0.512 | 0.512 | 0.512 | 0.512 |
| RSB-6 | 0.86 | 0.838 | 0.786 | 0.952 | 1.0 | 0.964 | 0.516 | 0.516 | 0.516 | 0.516 | 0.516 | 0.516 |
| RSB-7 | 0.802 | 0.832 | 0.766 | 0.906 | 0.94 | 1.0 | 0.496 | 0.496 | 0.496 | 0.496 | 0.496 | 0.496 |
| SP-5 | 0.462 | 0.46 | 0.456 | 0.458 | 0.46 | 0.462 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-10 | 0.514 | 0.504 | 0.504 | 0.532 | 0.536 | 0.522 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-25 | 0.472 | 0.502 | 0.498 | 0.472 | 0.466 | 0.462 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-50 | 0.45 | 0.49 | 0.462 | 0.484 | 0.47 | 0.468 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-100 | 0.462 | 0.504 | 0.518 | 0.508 | 0.498 | 0.486 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-1634 | 0.484 | 0.514 | 0.52 | 0.506 | 0.518 | 0.514 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |

Table 7: Cross-algorithm agreement with seeding according to an 50%-50% vote split, and 50% noise.

| | NC | GM | LP | RSB-5 | RSB-6 | RSB-7 | SP-5 | SP-10 | SP-25 | SP-50 | SP-100 | SP-1634 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NC | 1.0 | 0.672 | 0.672 | 0.672 | 0.672 | 0.672 | 0.328 | 0.328 | 0.328 | 0.328 | 0.328 | 0.328 |
| GM | 0.582 | 1.0 | 0.732 | 0.914 | 0.914 | 0.89 | 0.044 | 0.044 | 0.044 | 0.044 | 0.044 | 0.044 |
| LP | 0.526 | 0.998 | 1.0 | 1.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| RSB-5 | 0.576 | 0.904 | 0.76 | 1.0 | 0.988 | 0.984 | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 |
| RSB-6 | 0.518 | 0.904 | 0.732 | 0.984 | 1.0 | 0.992 | 0.054 | 0.054 | 0.054 | 0.054 | 0.054 | 0.054 |
| RSB-7 | 0.49 | 0.922 | 0.706 | 0.984 | 0.994 | 1.0 | 0.016 | 0.016 | 0.016 | 0.016 | 0.016 | 0.016 |
| SP-5 | 0.616 | 0.442 | 0.482 | 0.352 | 0.358 | 0.364 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-10 | 0.672 | 0.48 | 0.554 | 0.384 | 0.386 | 0.398 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-25 | 0.832 | 0.526 | 0.658 | 0.476 | 0.486 | 0.51 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-50 | 0.93 | 0.69 | 0.854 | 0.604 | 0.604 | 0.616 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-100 | 0.994 | 0.858 | 0.97 | 0.812 | 0.81 | 0.81 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SP-1634 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |

Table 8: Cross-algorithm agreement with seeding according to an 33%-66% vote split, and 50% noise.

Table 9: Cross-algorithm agreement with seeding according to an 80%-20% vote split, and 25% noise.