CSCI673 Final Project: Core & Community Detection using Justified Representation

Joonyoung (Aaron) Bae, Anish Jayant,Chandra Sekhar Mukherjee

Spring 2025

Abstract

Community Detection has been a central problem in graph theory and its practical applications can be found in various fields such as Clustering in Unsupervised Learning and RAG in Machine Learning. It has been observed that many simple community detection algorithms (like degree centrality) perform poorly when the communities have vastly different sizes, which has lead to recent interest and improvements with methods based on *balanced* centrality. A morally similar issue in voting theory occurs in committee selection, where slightly preferred candidates may disproportionately occupy committee chairs. Like relative centrality, the notion of Justified Representation (JR) was introduced to mitigate this problem. In this paper, we explore the wealth of JR algorithms applied to graph embeddings as a balanced/proportional centrality measure. Roughly, when edges are viewed as votes and committee size as the number of clusters, we observe that JR can give strong balanced centrality results.

Against this background, we investigate the merits of committee-selection methods with JR guarantees as centrality methods, by simulating their behavior in data with a single community, and then observe their effectiveness as *balanced centrality methods* in graphs with multiple underlying communities. We observe that "Method of Equal Shares (MES)" performs admirably as a balanced centrality method while ranking the core vertices (more separable nodes) at the top of the ranking, even outperforming recent relative-centrality-based measures. Despite the success, we warn that MES has a significantly higher run-time compared to other centrality methods.

Finally, we make an effort towards designing a clustering algorithm that aims to leverage these balanced-extracted cores to cluster the entire graph. However, we fall short of our goals in this step. We describe our approach in detail and discuss future directions for this part.

1 Introduction

Centrality Measure In network analysis literature *centrality measures* aim to quantify the importance or influence of individual nodes within a network. Among the most widely studied, ones mentioned in this paper are:

- **Degree Centrality.** The simplest measure, defined as the number of edges incident on a node. Nodes with high degree centrality have many direct connections and can act as local hubs.
- **Betweenness Centrality.** Counts the fraction of all-pairs shortest paths that pass through a node. Nodes with high betweenness often serve as critical bridges or bottlenecks [Fre77].
- **PageRank.** A variant of eigenvector centrality originally designed for ranking web pages. Each node distributes its score equally among its outgoing links, and scores are computed iteratively until convergence [BP98].

These measures provide complementary perspectives on node importance. Degree captures local reachability, betweenness identifies control points for information flow, and eigenvector-based methods such as PageRank incorporate the influence of neighbors' importance.

In general, the alignment of the context of "importance" and the underlying driving principles of centrality measures decides the usefulness of these methods for a graph. In this direction, an interesting question is to observe the behavior of centrality measures in graphs with underlying community structure.

Community Detection Community detection refers to the task of partitioning a graph into groups (or "communities") of nodes that are more densely connected internally than they are to nodes in other groups. It plays a central role in network analysis, as communities often correspond to functional units in social, biological, and technological systems. Key objectives include:

- **Modularity Maximization.** Identify a partition that maximizes the modularity score, which measures the difference between the observed intra-community edge density and the expected density under a random graph model. [NG04]
- **Stochastic Block Models.** Fit probabilistic generative models in which edges are drawn according to community-specific connection probabilities, and infer the latent labels via likelihood-based methods. [HLL83]
- **Spectral Clustering.** Use the eigenvectors of the graph Laplacian to embed nodes in a low-dimensional space where traditional clustering (e.g. k-means [Llo82]) can separate communities. [VL07]

Different algorithms trade off computational efficiency, resolution limits, and robustness to noise; selecting an appropriate method depends on the size, density, and expected structure of the network under study. In this empirical work, we will focus on normalized mutual information (NMI) and adjusted rand. index (ARI) to compare our results to the ground-truth cluster labelings. Observe that both measures, defined as follows, are permutation invariant. If Y is the labeling returned by the algorithm and C is the true classification, then

$$\mathrm{NMI}(Y, C) := \frac{2I(Y; C)}{H(Y) + H(C)}.$$

Note that large NMI (i.e., close to 1) implies that the output is strongly correlated with ground-truth labelings.

ARI is a more granular and based off of rand. index (RI), which operates on pairs of vertices. The RI measure counts the fraction of the $\binom{n}{2}$ pairs $i, j \in V$ that are either placed in the same cluster by both Y, C or different clusters by both Y, C (that is, Y, C agree on their inter-cluster status). The adjustment by ARI aims to "normalize" RI such that random assignments have $\mathrm{ARI} = 0$ and $\mathrm{ARI} \in [-1, 1]$ where better labelings have larger ARI.

$$\mathrm{ARI} := \frac{\mathrm{RI} - \mathbb{E}[\mathrm{RI}]}{\max(\mathrm{RI}) - \mathbb{E}[\mathrm{RI}]}$$

More concretely, if n_{ij} is the number of vertices predicted label i but actually j and $a_i = \sum_j n_{ij}$, $b_j = \sum_i n_{ij}$ are the marginals, then it is not hard to verify in closed form

$$\mathbb{E}[\mathrm{RI}] = \sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} / \binom{n}{2}, \mathrm{RI} = \sum_{ij} \binom{n_{ij}}{2}, \max(\mathrm{RI}) = \frac{1}{2}\left(\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right).$$

Clustering Using Community Detection Clustering algorithms aim to partition a set of points into groups such that intra-cluster similarity is maximized while inter-cluster similarity is minimized. There exist many clustering algorithms that given a vector data, create a graph with edges as an affinity matrix to find the best partition of the points. One example is clustering using Spectral Clustering in the popular Machine Learning library Scikit-learn, where they use RBF (Gaussian) Kernel or k-Nearest Neighbors to form the affinity matrix.

Throughout the paper, the terms clustering and community detection are often used interchangeably, as we follow the common practice of running community detection algorithms on a generated affinity matrix (for which we use k-NN graph in this paper), which often yields great performance [BHS+18] [VdMH08]

Utility of centrality measures in community detection Next, we discuss an interesting application of centrality measures in community detection. We have mentioned that centrality measures find "influential" parts of the network. In the graph embedding of vector datasets, it has been observed that the *high density regions* of the underlying data often leads to the vertices with high centrality score in methods such as degree-centrality and PageRank. Furthermore, in many cases, these central vertices may even be *more separable* into their ground truth clusters compared to the entire graph. This phenomenon was observed especially for graph embeddings of biological data in [MZ25].

Unbalancedness in traditional centrality measures The strategy proposed by [MZ25] is as follows. Let us suppose there is a graph with underlying community structure such that separating the entire graph into those true communities is very difficult. Instead, one may first try obtain the central nodes, and then cluster just those nodes. If these high centrality nodes both i) contain significant amount of vertices contain vertices from all underlying communities, and ii) are easier to separate than the entire graph, then one may end up with a better quality clustering of a subset of the points (vertices). This can still have many applications in biology.

However, [MZ25] observed that when the size/density-structure of different underlying clusters vary, the central nodes on the data's graph embedding may be *unbalanced*. That is, if we select the most central, say 20% of the vertices, it may not contain vertices from some underlying clusters at all.

To mitigate this, [MZ25] coined an approach called *relative centrality* that ranks the vertices in such a way that in many reasonable models (such as concentric stochastic block model or graph embedding of concentric GMM), the top-ranked vertices contain vertices from all communities, and this phenomena is also observed for real-world datasets.

In this direction, this project coins and investigate the following question.

Question 1 *Can we use voting algorithms as balanced centrality measures for graphs with underlying community structure?*

Especially, does the proportional fairness of voting algorithm translate to a balanced ranking in graphs with certain underlying structures?

Finally, can we leverage this centrality measure to obtain a better clustering of the whole graph? (In contrast to works like [MZ25] that only focuses on separating the high centrality nodes).

In this direction, we first define the concept of approval-based multi-winner selection.

1.1 Multi-winner selection

Approval-based multi-winner committee selection is a voting framework where each voter submits a list of approved candidates (rather than ranking or scoring them), and the task is to select a fixed-size committee that best represents these approvals. Unlike single-winner voting, this setting must account for diversity, proportionality, and representation across different voter groups. Various rules—such as Proportional Approval Voting (PAV), Sequential PAV, and the Monroe rule—capture different trade-offs between individual satisfaction and group fairness. This framework is central to applications like participatory budgeting, collaborative filtering, and representative decision-making, and raises deep research questions in algorithm design, complexity, and social choice theory.

Fairness is essential in approval-based multi-winner committee selection because it helps ensure that all groups of voters have a voice in the final outcome. If the process isn't fair, it can easily favor the majority and leave minority views out, which can make the results feel biased or unrepresentative. In this direction,

ideas like proportional and justified representation aim to capture what fairness should look like in these settings.

Justified Representation (JR) Different notions of Justified Representation (JR) are devised to capture increasingly strong guarantees of proportionality in approval-based multiwinner elections. Below are the standard definitions of JR, Proportional JR (PJR), and Extended JR (EJR) [Azi18]:

Let N be the set of voters, C the set of candidates, and $W \subseteq C$ a committee of size $|W| = k$. For each voter i, let $A_i \subseteq C$ be the approval set. (Note that EJR $\implies$ PJR $\implies$ JR)

$$\text{JR:} \quad \forall S \subseteq N,\ |S| \geq \frac{n}{k} \ \wedge \ \left|\bigcap_{i \in S} A_i\right| \geq 1 \quad \implies \quad \left|W \cap \bigcup_{i \in S} A_i\right| \geq 1$$

$$\text{PJR:} \quad \forall \ell \geq 1,\ \forall S \subseteq N,\ |S| \geq \ell \frac{n}{k} \ \wedge \ \left|\bigcap_{i \in S} A_i\right| \geq \ell \quad \implies \quad \left|W \cap \bigcup_{i \in S} A_i\right| \geq \ell$$

$$\text{EJR:} \quad \forall \ell \geq 1,\ \forall S \subseteq N,\ |S| \geq \ell \frac{n}{k} \ \wedge \ \left|\bigcap_{i \in S} A_i\right| \geq \ell \quad \implies \quad \exists\, i \in S :\ \left|W \cap A_i\right| \geq \ell$$

The intuition is that these gradually stronger conditions will guarantee more central nodes to be selected in the future sections. Next, we describe some popular committee election methods.

1.2 Committee Election (CE) Methods

Committee election or multiwinner voting refers to the electoral system that elects multiple candidates at once, with the provided preference from every voters [BDM98] [EFSS17]. In this project we focus on the approval ballot based committee election for simplicity. Throughout the paper, we mainly focus on three commmittee voting methods—Greedy-AV (AV), Sequential Phragmén (SP) and Method of Equal Sharees (MES)—that we have implemented based on the implementation given in [PVG+11]. These polynomial time algorithms were selected to see how different levels of JR affect the centrality results as AV does not satisfy any JR, SP satisfies JR and PJR, and MES satisfies JR, PJR, and EJR.

1.2.1 Greedy AV (AV)

The rule known as Greedy Approval Voting (AV) is nothing more than selecting the k candidates with the highest approval scores—in other words, multi-winner Approval Voting. This idea was first formalized in the classic paper [BF78]. In our later graph setting, this will just refer to selecting nodes with highest number of incoming edges. It is trivial that this simple greedy approach can leave a cohesive group without any representative if another separate cohesive group has a greater number of approvals.

1.2.2 Sequential Phragmén (SP)

Sequential Phragmén is an iterative load-balancing rule originally proposed by Edvard Phragmén [Phr96]. In each round, it selects the candidate whose addition to the current committee would lead to the smallest possible increase in the maximum "load" born by any voter. Concretely:

- Initialize each voter's load to zero.
- Repeat until k candidates are chosen:
    1. For each unelected candidate c, compute the minimal value ℓ_c such that, if c were elected and each approving voter's load increased uniformly to cover one seat, no voter's total load would exceed ℓ_c.

2. Elect the candidate c^* with the smallest ℓ_{c^*}.
3. Update each approving voter's load to reflect their share of the cost for c^*.

Because it spreads representation as evenly as possible, SP often achieves better proportionality than plain AV, and it satisfies Proportional Justified Representation (PJR) but generally not Extended Justified Representation (EJR) [BFJL24]. In our graph setting, SP corresponds to iteratively picking nodes so as to minimize the maximum "coverage load" on their neighbors.

1.2.3 Method of Equal Shares (MES)

The Method of Equal Shares, also known as Rule X, is a proportional approval-based rule. [PS20] [LS23]. It proceeds in two phases:

- **Phase 1 (Equal-Share allocation):**
  - Each voter is given an equal budget of $\frac{k}{n}$.
  - Repeatedly elect the candidate c with the smallest "price" q_c, where q_c is the minimum uniform per-voter contribution needed to fund c from among its approvers.
  - Deduct each approving voter's contribution $\min\{b_v, q_c\}$ from their remaining budget b_v.
  - Continue until no remaining candidate can be funded by its approvers.
- **Phase 2 (Completion):** If fewer than k candidates have been elected, fill the remaining seats using a completion rule (e.g. Sequential Phragmén) on the residual budgets and loads.

MES guarantees Proportional Justified Representation (PJR) and Extended Justified Representation (EJR) [PS20]. In our graph setting, MES corresponds to buying nodes at increasing per-voter "prices," ensuring that dense, cohesive neighborhoods receive representation before sparser ones.

2 Committee Election as a *balanced* Centrality Measure on the Graph

Voting algorithms for committee election aims to choose a committee that in some notion well-represents the choice of a population. In this direction, a lot of work has taken place in coming up with committee election algorithms that can select a *fair* committee. In this project, we sought to uncover the connection of such algorithms with community detection in graphs. We are motivated by the following paper [MZ25] that observed in various biology datasets, the communities portray a structure called core-periphery, where each community has a dense central region and a sparser region around it so that the community overlaps happen in these sparser regions. The authors showed that if one obtains the k-nearest-neighbor embedding of datasets with underlying clusters, the resultant graph has Multi-Core-Periphery with Communities(MCPC) structure. Under MCPC, edges from more central nodes (core nodes) end up in the same core with high probability and densely connected, while the peripheries are loosely connected,with inter-community edges originating more frequently in the peripheries. Throughout this paper, we also leverage this observation and use similar assumption for the success of Committee Election based centrality measures, since we build our intuition upon the assumption that each cluster has a well-separated and dense core, which can be recovered by approval ballot based committee election rules.

Can committee election algorithms with fairness guarantees be used to aid clustering? We explore this question with a thorough investigation of committee election algorithm performance on different kinds of graph embeddings of vector datasets (including ones with underlying MCPC structure) and observe that these methods indeed perform better than traditional centrality measures in recovering better separable parts of the underlying communities, often matching the performance of relative centrality-based approach.

We further aim to utilize this phenomena to obtain a better clustering of the entire dataset, but run into some trouble here, which we describe and leave open as future direction.

We begin our project by the following statement:

Claim: committee election (CE) methods such as Sequential Phragmén (SP) and Method of Equal Shares (MES) used on a graph can find highly central core nodes that are more separable in a balanced manner, where the number of core nodes selected from each community is proportional to the size of each community. Lastly, compared to other centrality measure algorithms, these voting rule based centrality measures are robust to change in the parameter.

The following sections delve deeper into the aforementioned claims one by one and provides insights with experimental results and intuitions behind them.

2.1 Committee Election based Centrality Measure

Our approach leverages from the novel perspective on a graph structure. Let's view a directed graph (or undirected graph with every edge as a bi-directed edge) in the following manner:

- The main idea is to think of nodes V as voters N and edges $(i, j) \in E$ in any graph as i's approval of j. For simplicity, we will consider an undirected graph with (i, j) meaning an approval in both ways.
- Also with the same reason, we will first say everyone is both a voter and a candidate with no self approval to avoid potential problems.
- To capture more context of the graph, we define each voter v_i's approval profile $A_i \subseteq V$, which is the set of candidates approved by v_i, as set of nodes within h-hop distance from v_i. The effect of this h-hop parameter is explored in this section.
- With above settings, we run committee election and show the selected committee nodes are the nodes with high centrality, often forming more clusterable subsets of the nodes.

  As shown in Figure 1, in a simple setting with one community following a Gaussian distribution, the three listed committee election methods [AV, SP, MES] can find the central points of the community. The intuition is that more dense regions or the nodes with more incoming edges will be selected earlier.

  We also have tested with different k values for k-nearest-neighbor search that was used to build the edges from a vector dataset. As mentioned, this k value is analogous to h in h-hop in the community setting. Overall, this parameter sets how far each node "approves" and the intuition is that this is the parameter of how local or global we want to explore when we calculate the core nodes. As shown in figure 1, as this k value decreases, more local core nodes are selected, while large k value concentrates the selected nodes to more global core nodes.

  It is to be noted that there exists limitation to the SP or MES to be used as the centrality measure on all the nodes. The primary bottleneck is that since these algorithms are sequential, the centrality scores vary every iteration after one committee member was elected and the values of each node has been updated. Thus, we have to perform the sequential algorithm until all the nodes have been elected, which leads to a quadratic (in number of points) run-time making it impractical for large graphs. In fact, the naive run-time (as in the abcvotingrule package) to get the top x-ranked nodes is $\Omega(x \cdot |E|)$, although we believe this can be improved.

  Furthermore, for MES there maybe multiple vertices (candidates) in a single-round that satisfy the selection criteria, and the order in which they are selected may affect the final outcome.

2.2 Balancedness of CE-based Centrality Measure

The balancedness of centrality measure can be very useful as previously mentioned. In this section, we show in Figure 2 that CE-based centrality measure, specifically on SP and MES, show empirical evidence

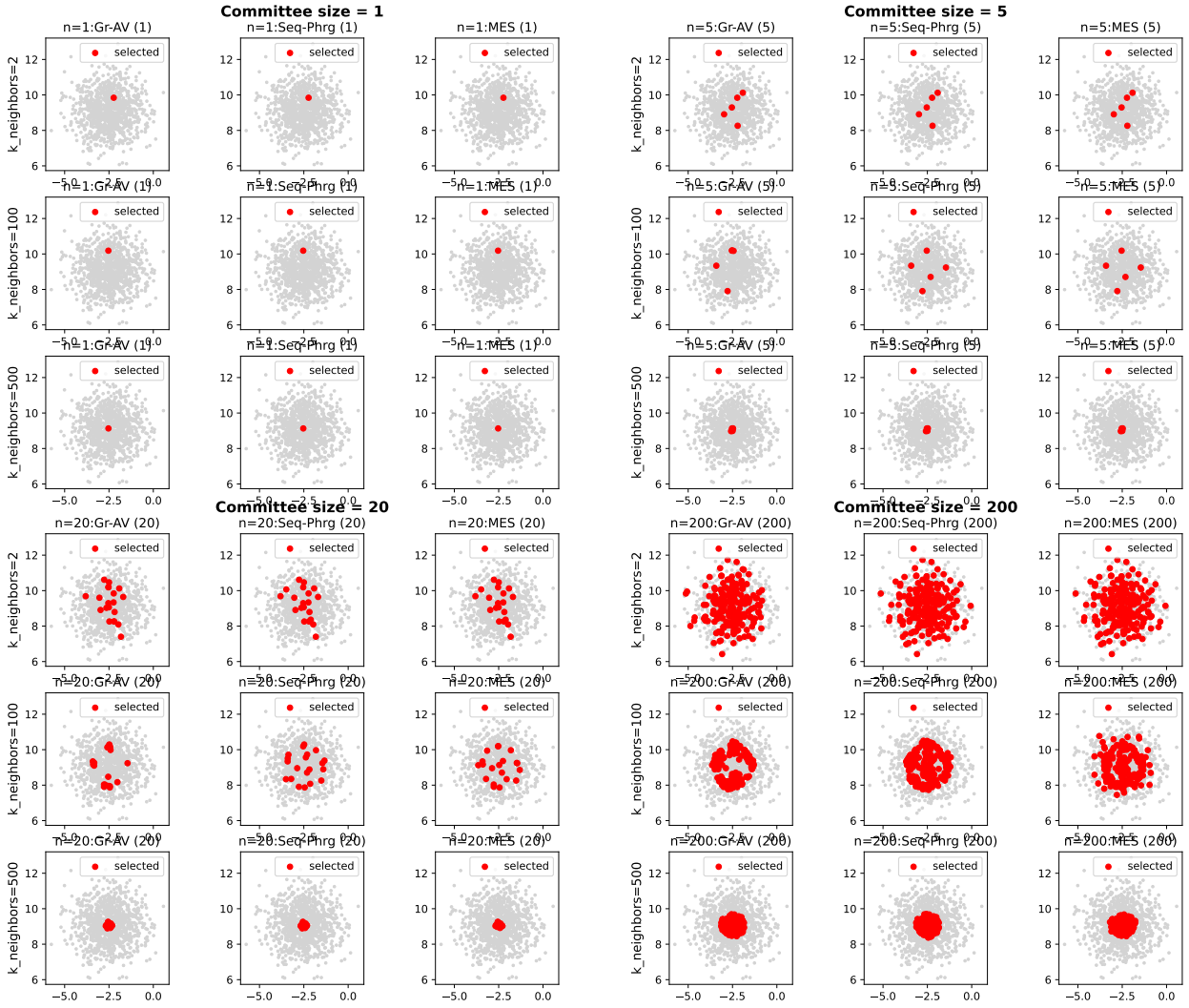


Figure 1: We first generate a single cluster model with 1000 datapoints, where the datapoints are samples from a Gaussian Distribution in $\mathbb{R}^2$. Then we build k-Nearest-Neighbor graph for k = (2, 100, 500) and run 3 different CE algorithms (AV ,SP, MES) with 4 different committee sizes (1, 5, 20 ,200). The selected committee nodes are highlighted in red. It is shown that as k value increases, the centrality measure also shifts from local to more global context, "converging" to (0, 0). Also, it is shown that all three algorithms are capable of finding core nodes in this simple setting.

of balancedness, successfully recovering core nodes proportional to community sizes, even from the smallest community in the unbalanced dataset. Another crucial observation is that the core nodes selected are more separated and thus more clusterable, which lays the groundwork for the clustering explored in the later section.

Same experiment with MES without completion using SP showed that only small number of core nodes is recovered due to the harsh constraint, but the selected core nodes are still central and of high accuracy.

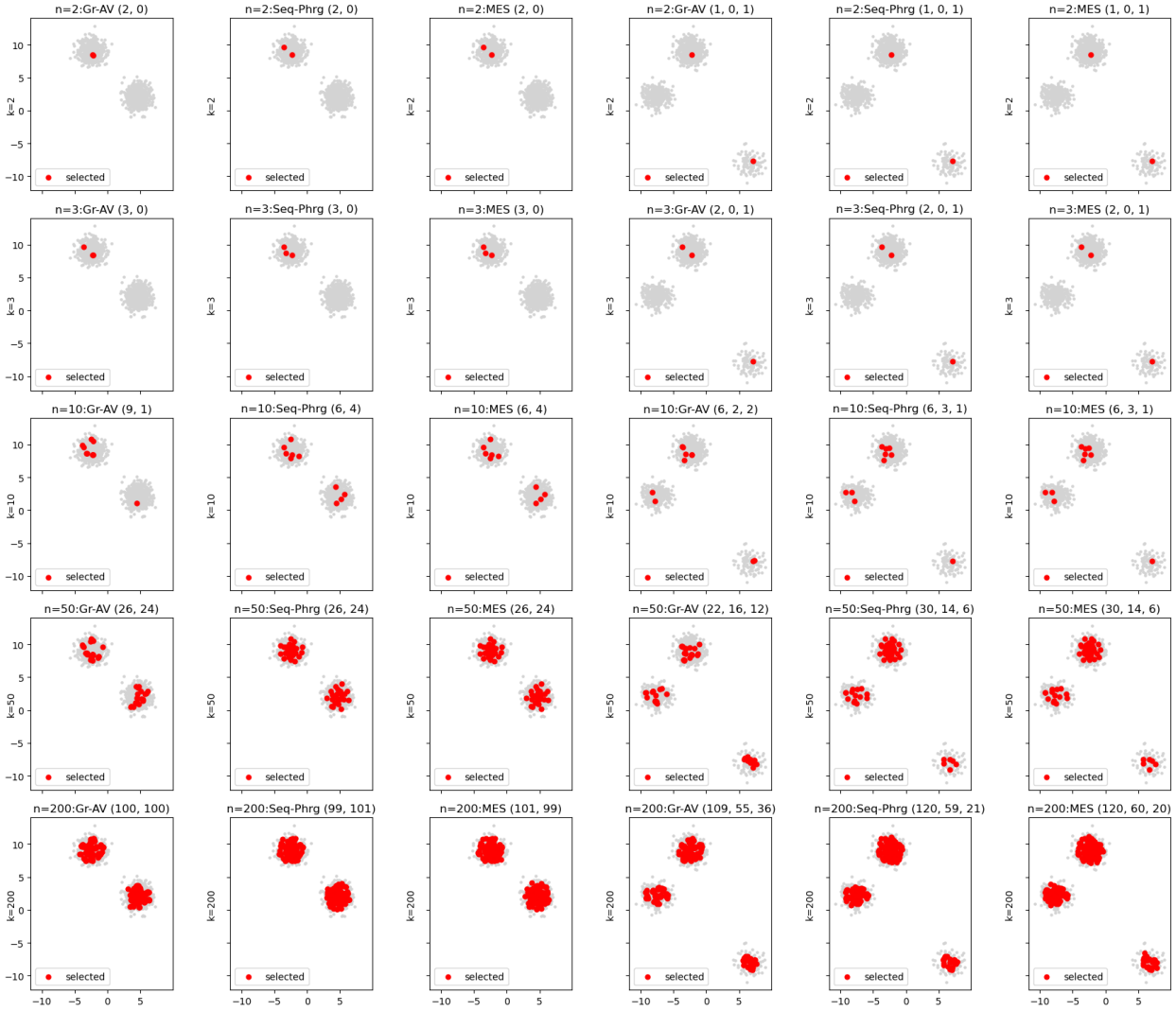
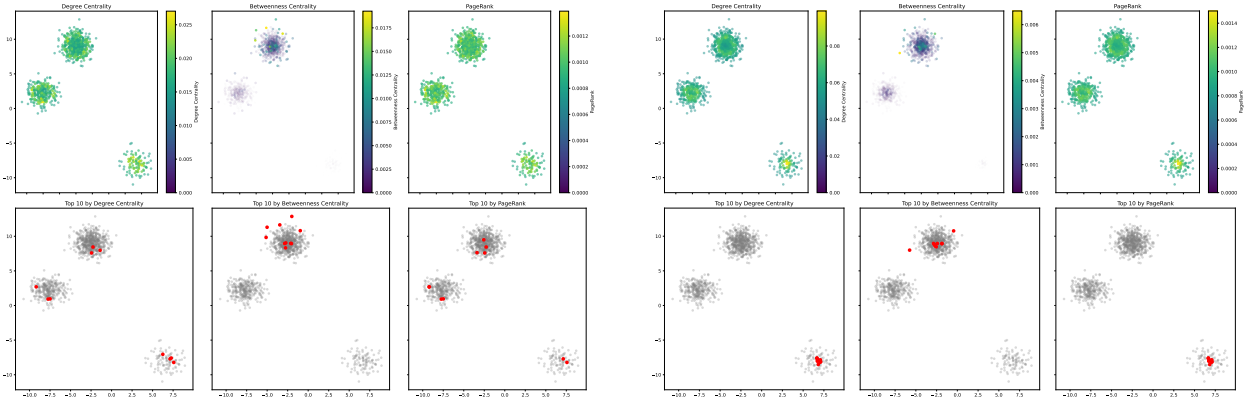


Figure 2: The dataset on the left is 1000 datapoints sampled from two identical gaussian distributions (500 each) and one on the right is 1000 datapoints sampled from 3 identical gaussian distributions (600, 300, 100 datapoints each). We created directed edges by finding 15 Nearest Neighbors V_i from each node i and connecting $i \to V_i$. The central core points found by SP and MES showed strong balancedness in both datasets, while AV returned highly varying core points, since AV only returns the highest-density core nodes. One such occasion can be seen at (n=10) for AV.

2.3 Robustness to K-NN Parameter

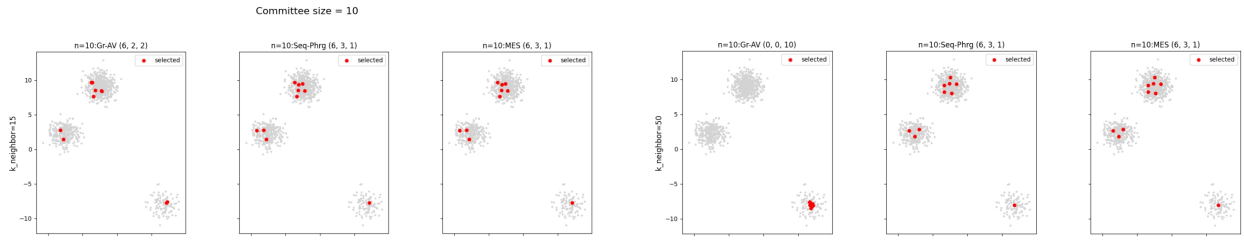
In this section, we show that SP and MES based centrality are robust to k in k-NN (h in h-hop) parameter selection compared to other popular centrality measures. As shown in Figure 3 and 4, Degree Centrality,

Betweenness, PageRank, and AV showed high sensitivity to k parameter in k-NN graph generation, while SP and MES centrality showed robustness to change in the parameter. This is due to the mechanism both methods utilize in assuring that once a node is selected, nearby nodes are discouraged to be selected in the coming iterations, allowing farther away points to be selected next.



(a) $k = 15$. (b) $k = 50$.

Figure 3: Comparison of centrality measures (Degree, Betweenness, PageRank) on the unbalanced 3-GMM dataset (from Figure 2) for two different KNN sizes. (a) $k = 15$, (b) $k = 50$.
Nodes with top 10 centrality scores are highlighted in red. Given the true proportion of cluster sizes are (6,3,1), When $k = 15$, Degree finds (3,3,4), Betweenness finds (10,0,0) and Pagerank finds (4,4,2), while when $k = 50$, all methods find all top 10 nodes in one cluster, showing the sensitivity of the performance to the k parameter in k-NN. The top row shows the heatmap of the centrality score on all datapoints.



(a) $k = 15$. (b) $k = 50$.

Figure 4: Robustness of SP and MES based centrality measures across two KNN sizes: (a) $k = 15$, (b) $k = 50$. is shown here since they were able to find top 10 nodes with the correct ratio of (6,3,1) compared to the size of each cluster. However, AV has shown high sensitivity to k parameter in k-NN graph generation, failing to find balanced core points when $k = 50$.

3 Part 2: Clustering/Community Detection using CE

In the previous part, we have observed that the PV and MES can find balanced core nodes from a density perspective in well-separated 2-dimensional Gaussian distribution, robust to the parameter k used in the embedding. In this section, we test the balancedness and ability to recover cores in harder-to-cluster scenarios. To this end, we use an overlapping two-cluster-concentric GMM [R+09] setting, defined in the next section, following [MZ25]. This model appears intimately related to many natural data distributions.

We observe that MES find core nodes that are balanced and more separate. Noticeably, the balancedness obtained by MES is even better than N-Rank, the simple method proposed by [MZ25].

3.1 Concentric GMM Model

We test our hypothesis of balancedness and clusterability of the core nodes selected by MES on the noisy and overlapping concentric GMM model, which is much harder model than already well-separated models used in the previous sections.

Data generation. We consider two centers $c_\ell \in \mathbb{R}^d$, for $\ell \in \{0,1\}$. Around each center c_ℓ, we place two isotropic Gaussian clusters:

$$D_{\ell,1} = \mathcal{N}\big(c_\ell,\ \sigma_{\ell,1}^2\, I_d\big), \qquad D_{\ell,0} = \mathcal{N}\big(c_\ell,\ \sigma_{\ell,0}^2\, I_d\big),$$

with variances satisfying $\sigma_{\ell,0} \geq 1.1\,\sigma_{\ell,1}$. From each distribution $D_{\ell,j}$, we sample $n_{\ell,j}$ points, which we denote by

$$V_{\ell,j} = \{\, x_1^{(\ell,j)}, \ldots, x_{n_{\ell,j}}^{(\ell,j)} \,\}.$$

Thus the two underlying communities are

$$V_\ell = V_{\ell,1} \cup V_{\ell,0}, \quad \ell \in \{0,1\}.$$

In our simulations we set

$$d = 20, \qquad n_{\ell,j} = 100 \quad \text{for all } \ell \in \{0,1\},\ j \in \{0,1\},$$

and choose variances

$$(\sigma_{0,1},\, \sigma_{0,0},\, \sigma_{1,1},\, \sigma_{1,0}) \;=\; \big(0.1,\ 0.3,\ \gamma \cdot 0.1,\ \gamma \cdot 0.3\big), \qquad 1 \leq \gamma < 3.$$

The two centers c_0 and c_1 are placed close enough so that the Gaussians overlap. Finally, given the combined dataset $V_0 \cup V_1$, we compute the 20-nearest-neighbor graph and its embedding to get a directed graph.

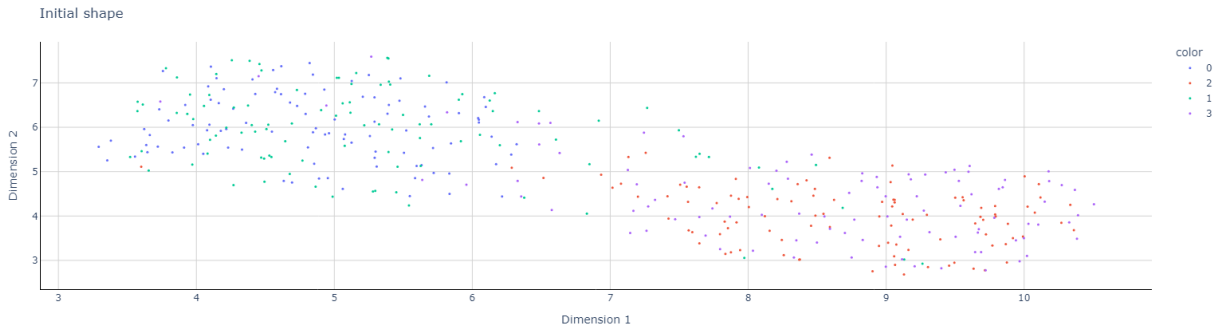


Figure 5: The 2-dimensional UMAP [MHM18] embedding of 20-dimensional 2-clusters Concentric GMM Model generated with the above model definition and hyperparameters (with $\gamma = 1$) is shown above. This model conveys MCPC structure with overlapping peripheries that make any clustering algorithms such as K-Means harder to cluster. Each cluster conveys dense and farther away cores, which might be less evident due to projection down onto 2-dimensional space.

We show the results in Table 1 with the comparison of other centrality measure: Degree Centrality and N-Rank [MZ25]. MES was used as the CE algorithm since SP's runtime is too slow to be practically used on a large-scale data. Even MES takes more than 10 minutes for finding top 100 nodes in 1,000 nodes GMM model, and finding from 10,000 nodes is expected to take more than an hour to run. It is shown that MES can find more balanced and clusterable core nodes, as shown by analyzing the simple K-Means results on the core nodes found.

γ	Method	NMI	ARI	Accuracy	Balancedness
1.00	All points	0.758	0.846	0.960	**1.000**
	MES	**1.000**	**1.000**	**1.000**	**1.000**
	Degree	**1.000**	**1.000**	**1.000**	0.975
	N-Rank	**1.000**	**1.000**	**1.000**	0.975
1.25	All points	0.635	0.739	0.930	1
	MES	**0.854**	**0.900**	**0.975**	0.925
	Degree	0.756	0.805	0.950	0.925
	N-Rank	0.761	0.805	0.950	**0.975**
1.50	All points	0.541	0.647	0.902	1
	MES	**0.842**	**0.899**	**0.975**	**0.825**
	Degree	0.816	0.889	**0.975**	0.725
	N-Rank	0.838	0.897	**0.975**	0.800
2.00	All points	0.564	0.655	0.905	1
	MES	**0.842**	**0.899**	**0.975**	**0.825**
	Degree	0.748	0.847	**0.975**	0.625
	N-Rank	0.816	0.889	**0.975**	0.725

Table 1: KMeans clustering results on all points and on top 10% core nodes selected by various centrality measures. NMI, ARI, Accuracy, and balancedness for varying γ are shown. Higher γ indicates that one of the clusters have higher variance, leading to an unbalanced multi-core-periphery structure. The best values in each column for a fixed γ are shown in **bold**.

3.2 Attempts at translating the extraction of better separable cores to a better clustering algorithm on the whole dataset

We return from the successes of finding balanced cores in the previous section and revisit our chief objective of clustering the *entire* graph. Throughout the experiments, AV was too unstable to be used as a balanced centrality measure, SP ran too slow to be used in practice. However, MES shows potential with tractable runtime for small datasets and strong performance for finding balanced cores compared Degree and N-Rank, as shown in the previous section. Thus, we devised a new clustering algorithm that aims to cluster all the datapoints (not just the top centrality score nodes) leveraging from the initial core set found by MES. We show our novel algorithm in Algorithm 1. The visual representation of the algorithm is shown in Figure 6. The results of our new Algorithm is then compared to K-Means in Table 2. The suboptimal results show that there are further room for improvement.

Description of Algorithm To cluster a dataset X of n elements into K clusters, we first create a 20-nearest-neighbors graph embedding G. Then, for a fixed constant p, select a p-fraction of vertices from G as the core set, using MES. By running k-means with $k = K$ on this core set, we label every $c \in C$ as one of $[K]$. It remains to propagate these labels to the unlabeled set $X \backslash C$, which is done iteratively until $C = X$. At each iteration, each core vertex $c \in C$ with label $\ell(c) \in [K]$ votes for its label on the closest unlabeled vertex. Then, the unlabeled vertices which have received at least one vote adopt their respective highest-voted label and join the core set for the next iteration.

Algorithm 1 MES-Propagation-Clustering(X, p, K)

Require:
- $X = \{x_1, \ldots, x_n\}$: dataset of n points
- $p \in (0, 1]$: fraction of initial core nodes (default 0.1)
- K: number of clusters

Ensure: Assignment of each x_i to one of K clusters

Build the 20-nearest-neighbor graph G on X
$m \leftarrow \lceil p\, n \rceil$
Elect an initial core set $C \subseteq X$, $|C| = m$, via MES on approval ballots from G
Initialize label array $L[\,\cdot\,] \leftarrow$ `undef`
Run K-Means on $\{x : x \in C\}$ into K clusters; set $L[x] \leftarrow$ its cluster index
while $|C| < n$ **do**
    $\mathcal{U} \leftarrow X \setminus C$ ▷ the unlabeled points
    Initialize votes $v[u] \leftarrow \mathbf{0} \in \mathbb{Z}^K$ for each $u \in \mathcal{U}$
    for all $c \in C$ **do**
        $u^\star \leftarrow \arg\min_{u \in \mathcal{U}} \|c - u\|$ ▷ nearest unlabeled to c
        $v[u^\star][\,L[c]\,] \leftarrow v[u^\star][\,L[c]\,] + 1$
    end for
    for all $u \in \mathcal{U}$ **do**
        if $\max_k v[u][k] > 0$ **then** ▷ if received any vote
            $L[u] \leftarrow \arg\max_k v[u][k]$ ▷ new label as the highest voted label
            $C \leftarrow C \cup \{u\}$
        end if
    end for
end while
return L

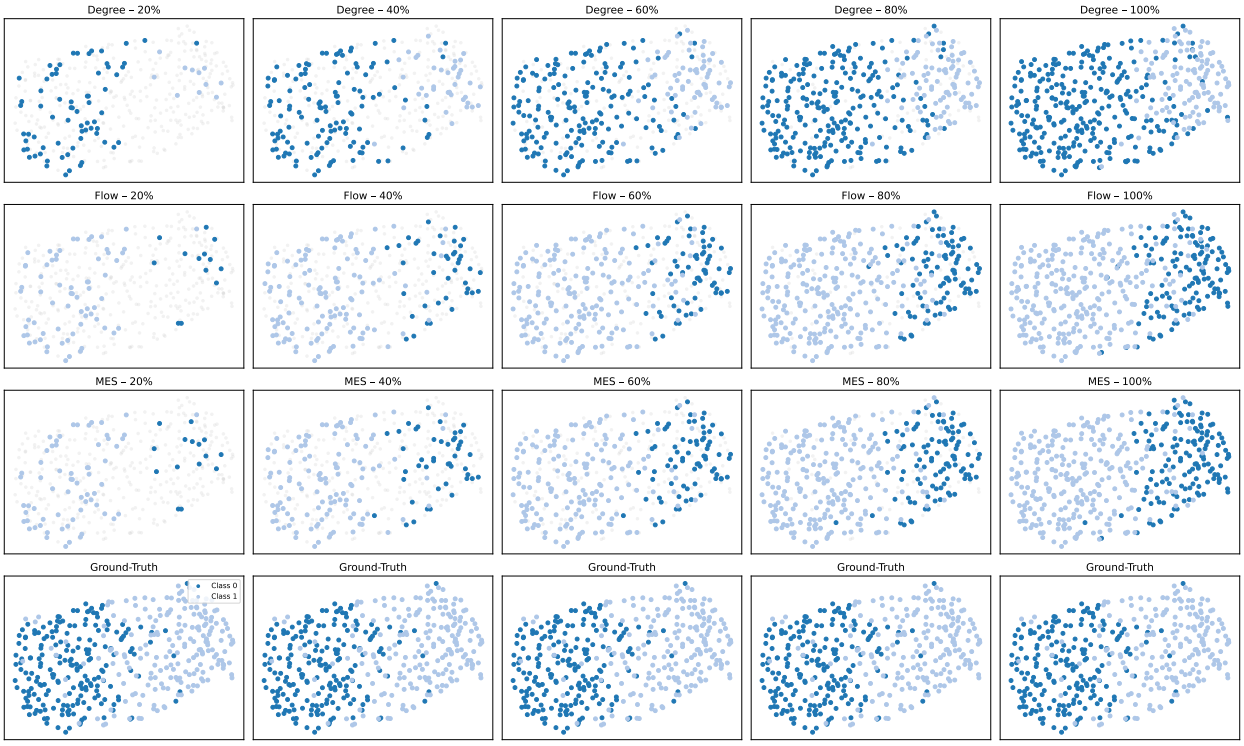


Figure 6: Clustering Results of our new clustering algorithm written in Algorithm 1. We show the visual representation of how the first set of core nodes are selected and labels then propagate to the nearest unlabeled nodes on a majority vote base. It is shown that all the algorithms suffered from the unbalancedness and high noise from the Concentric GMM with $\gamma = 1.5$, resulting in unbalanced clustering results then vanilla K-Means on the whole data. The fourth row shows the Ground-Truth labels of the concentric GMM model.

Method	NMI	ARI	Accuracy
K-Means	**0.541**	**0.647**	**0.902**
Degree	0.359	0.312	0.780
Flow	0.349	0.347	0.795
MES	0.416	0.421	0.825

Table 2: We show the results of the clustering on all datapoints of the above concentric GMM model with $\gamma = 1.5$. With that γ value, now one cluster has high variance, making the clustering task harder to achieve. First row shows the K-Means result on the whole data, which is the best among all the other algorithms. Three different centrality methods were used to identify the top 10% core nodes, which were then clustered by K-Means. Then labels of the cores were propagated using Algorithm 1.

4 Future Direction

The principle objective after finding cluster cores is to return to the entire data by propagating cluster labels. Indeed, the benefit of having *balanced* centrality is to ensure that, in this propagation process, all clusters have fair representation. Currently, the method we propose to diffuse labels as Algorithm 1 seems to lose the insights of balanced centrality, and thus performs worse than vanilla K-means (cf. **??**). In the current implementation, at each iteration, every core member $c \in C$ gives only the closest unlabeled point a "vote," (line 11) and the unlabeled point is classified after just a single vote (line 14). We posit that this propagation can be improved by (1) having each core member submit a ranking over $\mathcal{U}$ and (2) deferring classification of $u \in \mathcal{U}$ until some more robust property of $v[u]$ emerges (e.g., number of top-r rank votes) or an iteration limit.

Under the MCPC framework, it is likely that Algorithm 1 incorrectly classifies periphery nodes (the higher variance cluster in GMM) in early iterations and thereafter propagates these errors through periphery-periphery connections. By reducing the greedy-ness in this modification, we expect to usually delay periphery decisions, giving time for cores to fully develop, as they will receive very highly ranked votes if beginning from balanced cores. Furthermore, peripheries are likely to give their periphery neighbors lower ranked votes, thus containing the spread. Together, these observations ensure that labeling "travels" along core-periphery edges for as long as possible. An important idea here (one that we are actively working on) is to rather identify multiple layers in each community from the most core to most periphery, and propagate the labels layer-wise.

We conclude with a discussion on several future theoretical and algorithm-design questions that we plan to investigate.

1. Proving any correctness in our framework has generally proven difficult so far, given the short time-frame of the project. Here, the first thing we would like to prove is whether the top points selected by MES leads to high core prioritization (core nodes are selected above periphery nodes) and balancedness?

2. Currently, the run-time of MES seems to be $n \cdot x$ where x is the number of points selected. If the graph is sparse, (number of approval per person is few), then selecting one person (node) should not affect the structure of the most of the graphs. Could we theoretically prove that a significantly faster implementation exists that has run time linear in n and the multiplicative factor on x is logarithmic? The potential of MES in real-world datasets can be more thoroughly investigated if we are able to improve run time.

3. Regarding the previous point, an alternative approach is to select only a few points using the committee selection algorithm, and then propagate these ranking using something akin to personalized PageRank to obtain a faster algorithm.

References

[Azi18] Haris Aziz. Proportional representation in approval-based committee voting and beyond. *arXiv preprint arXiv:1802.00882*, 2018.

[BDM98] Hans-Hermann Bock, William HE Day, and FR McMorris. Consensus rules for committee elections. *Mathematical Social Sciences*, 35(3):219–232, 1998.

[BF78] Steven J Brams and Peter C Fishburn. Approval voting. *American Political Science Review*, 72(3):831–847, 1978.

[BFJL24] Markus Brill, Rupert Freeman, Svante Janson, and Martin Lackner. Phragmén's voting methods and justified representation. *Mathematical programming*, 203(1):47–76, 2024.

[BHS+18] Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology*, 36(5):411–420, 2018.

[BP98] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[EFSS17] Edith Elkind, Piotr Faliszewski, Piotr Skowron, and Arkadii Slinko. Properties of multiwinner voting rules. *Social Choice and Welfare*, 48:599–632, 2017.

[Fre77] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[HLL83] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

[Llo82] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[LS23] Martin Lackner and Piotr Skowron. *Multi-winner voting with approval preferences*. Springer Nature, 2023.

[MHM18] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[MZ25] Chandra Sekhar Mukherjee and Jiapeng Zhang. Balanced ranking with relative centrality: A multi-core periphery perspective. In *ICLR*, 2025.

[NG04] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[Phr96] Edvard Phragmén. Sur la théorie des élections multiples. *Öfversigt af Kongliga Vetenskaps-Akademiens Förhandlingar*, 53:181–191, 1896.

[PS20] Dominik Peters and Piotr Skowron. Proportionality and the limits of welfarism. In *Proceedings of the 21st ACM Conference on Economics and Computation*, pages 793–794, 2020.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[R+09] Douglas A Reynolds et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663):3, 2009.

[VdMH08] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[VL07] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007.