DISSECTING GRAPH MEASURES PERFORMANCE FOR NODE CLUSTERING IN LFR PARAMETER SPACE

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph measures can be used for graph node clustering using metric clustering algorithms. There are multiple measures applicable to this task, and which one performs better is an open question. We study the performance of 25 graph measures on generated graphs with different parameters. While usually measure comparisons are limited to general measure ranking on a particular dataset, we aim to explore the performance of various measures depending on graph features. Using an LFR generator of graphs with preferential attachment, we create a dataset of $\sim$7500 graphs covering the whole LFR parameter space. For each graph, we assess the quality of clustering with k-means algorithm for every considered measure. We determine the best measure for every area of the parameter space. We find that the parameter space consists of distinct zones where one particular measure is the best. We analyze the geometry of the resulting zones and describe it with simple criteria. Given particular graph parameters, this allows us to choose the best measure to use for clustering.

1 INTRODUCTION

Graph node clustering is one of the central tasks in graph structure analysis. It provides a partition of nodes into disjoint clusters, which are groups of nodes that are characterized by strong mutual connections. It can be of practical use for graphs representing real-life systems, such as social networks or industrial processes. Clustering allows to infer some information about the system: the nodes from one cluster are highly similar, while the nodes from different clusters are dissimilar. The technique can be applied without any labeled data to extract important insights about a network.

There are different approaches to clustering, including ones based on modularity optimization (Newman & Girvan, 2004; Blondel et al., 2008), label propagation algorithm (Raghavan et al., 2007; Barber & Clark, 2009), Markov cluster process (Van Dongen, 2000; Enright et al., 2002), and spectral clustering (Von Luxburg, 2007). In this work, we use a different approach based on defining a similarity measure on a graph, which allows one to use any metric clustering algorithm (e.g. Yen et al., 2009).

The choice of the measure significantly affects the quality of clustering. Classical measures are the *Shortest Path* (Buckley & Harary, 1990) and the *Commute Time* (Göbel & Jagers, 1974) distances. The former is the minimum number of edges in a path between a given pair of nodes. The latter is the expected number of steps from one node to the other and back in a random walks on the graph. There is a number of other measures, including recent ones (e.g. Estrada & Silver, 2017; Jacobsen & Tien, 2018), many of them are parametric. Despite that graph measures are compatible with any metric algorithm, in this paper we restrict ourselves to the kernel k-means algorithm (e.g. Fouss et al., 2016).

We base our research on a generated set of graphs. There are various algorithms to generate graphs with community structures. The well-known ones are the Stochastic Block Model (Holland et al., 1983) and Lancichinetti–Fortunato–Radicchi benchmark (Lancichinetti et al., 2008) (hereafter, LFR). The first one is an extension of the Erdős–Rényi model with different intra- and inter-cluster probablilities of edge creation. The second one is a development of preferential attachment ideas started with Barabási–Albert model (Barabási & Albert, 1999). There are other generation models, e.g., Naive Scale-free Clustering (Pasta & Zaidi, 2017). We choose LFR model—classical preferential attachment model, which generates graphs similar to real world ones.

There are a lot of measure benchmarking studies considering node classification and clustering for both generated graphs and datasets (Sommer et al., 2016; 2017; Avrachenkov et al., 2017; Ivashkin & Chebotarev, 2016), etc. Unlike these studies concluding with a global leaderboard, we are looking for the leading measures for each set of LFR parameters.

We aim to explore the performance of 25 selected measures in graph clustering problem on a set of generated graphs with different parameters. We assess the quality of clustering with every considered measure and determine the best measure for every region of the graph parameter space.

Our contributions are summarized as follows:

- We generate a dataset of $\sim$7500 graphs covering all parameter space of LFR generator;
- We consider a broad set of measures and rank measures by clustering performance on this dataset;
- We find regions of certain measure leadership in the graph parameter space;
- We determine the graph features that are responsible for measure leadership;

Our framework for clustering with graph measures as well as a collected dataset are available on `link_is_not_available_during_blind_review`.

2 DEFINITIONS

2.1 KERNEL k-MEANS

The original k-means algorithm (Lloyd, 1982; MacQueen et al., 1967) clusters objects in Euclidean space. It requires coordinates of the objects to determine the distances between them and centroids. The algorithm can be generalized to use the degree of closeness between the objects without defining a particular space. This technique is called *the kernel trick*, usually it is used to bring non-linearity to linear algorithms. The algorithm that uses the kernel trick is called kernel k-means (see, e.g., Fouss et al., 2016). For graph node clustering scenario, we can use graph measures as kernels for the kernel k-means.

Initially, the number of clusters is known and we need to set initial state of centroids. The results of the clustering with k-means are very sensitive to it. Usually, the algorithm runs several times with different initial states (trials) and chooses the best trial. There are different approaches to the initialization, we consider three of them:

- *Random data points*. For each cluster, pick one random object as the initial centroid.
- *k-means++*. For each cluster, one object is sequentially selected as the centroid. For the first cluster, a random object is selected. For the next cluster, the most distant object from previously selected centroids is picked (Arthur & Vassilvitskii, 2006).
- *Random partition*. The dataset is randomly split into clusters, then for each resulting cluster the centroid is calculated as the center of the cluster.

We combine all these strategies to reduce the impact of the initialization strategy on the result.

There are different criteria for choosing the best trial. The classical criterion for k-means is *inertia*, which is the sum of all distances from the labeled object to the centroid of the corresponding cluster (also known as *within-cluster sum of distances*). Since we are working with graphs, we can also use any unsupervised measure of graph partition quality. We use a well-proven *modularity* measure (Newman & Girvan, 2004) as such.

2.2 SIMILARITY MEASURES

For a given graph G, $V(G)$ is the set of its vertices and A is its adjacency matrix. A *measure* on G is a function $\kappa\colon V(G) \times V(G) \to \mathbb{R}$, which gets two nodes and returns similarity (bigger means closer) or dissimilarity (bigger means farther). Distances are dissimilarity measures.

A *kernel on a graph* is a graph nodes similarity measure that has an inner product representation. Any symmetric positive semidefinite matrix is an inner product matrix (also called Gram matrix). A kernel matrix K is a square matrix that contains similarities for all pairs of nodes in a graph.

To use kernel k-means, we need kernels. Despite that not all similarity measures we consider are Gram matrices, we treat them as kernels. For the list of measures bellow, we use the word "kernel" only for the measures that satisfy the kernel definition.

Classical measures are *Shortest Path* distance (Buckley & Harary, 1990) (SP) and *Commute Time* distance (Göbel & Jagers, 1974) (CT). SP is the minimum number of edges in a path between a given pair of nodes. CT is the expected lengths of random walks between two nodes. The CT distance is known to equal the *resistance distance* (Klein & Randić, 1993; Doyle & Snell, 1984; Bapat, 1999; Sharpe, 1967; Moore & Subak-Sharpe, 1968) up to a constant factor (Chandra et al., 1996).

SP and CT are defined as distances, so we need to transform them into similarities to use as kernels. We use the following distance to similarity measure transformation (Chebotarev & Shamis, 1998a; Borg & Groenen, 2005):

$$K = -H\mathcal{D}H;\ H = I - E/n \tag{1}$$

where $\mathcal{D}$ is a distance matrix, E is the matrix of ones, I is the identity matrix, and n is the number of nodes.

In this paper, we examine 25 graph measures. We present these measures grouped by type similarly to (Avrachenkov et al., 2017):

- Adjacency Matrix A based kernels and measures.
  - *Katz kernel*: $K_\alpha^{\text{Katz}} = (I - \alpha A)^{-1}$, $0 < \alpha < \rho^{-1}$, where ρ is the spectral radius of A. (Katz, 1953) (also known as Walk proximity (Chebotarev & Shamis, 1998b) or von Neumann diffusion kernel (Kandola et al., 2003; Shawe-Taylor et al., 2004))
  - *Communicability kernel* $K_t^{\text{Comm}} = \operatorname{expm}(tA)$, $t > 0$. expm means matrix exponential. (Fouss et al., 2006; Estrada & Hatano, 2007; 2008)
  - *Double Factorial similarity*: $K_t^{\text{DF}} = \sum_{k=0}^{\inf} \frac{t^k}{k!!} A^k$, $t > 0$ (Estrada & Silver, 2017).
- Laplacian Matrix $L = D - A$ based kernels and measures. $D = \operatorname{Diag}(A \cdot \mathbf{1})$ is the degree matrix of G, $\operatorname{Diag}(\mathbf{x})$ is the diagonal matrix with vector $\mathbf{x}$ on the main diagonal.
  - *Forest kernel*: $K_t^{\text{For}} = (I + tL)^{-1}$, $t > 0$ (also known as Regularized Laplacian kernel) (Chebotarev & Shamis, 1995).
  - *Heat kernel*: $K_t^{\text{Heat}} = \operatorname{expm}(-tL)$, $t > 0$ (Chung & Yau, 1998).
  - *Normalized Heat kernel*: $K_t^{\text{NHeat}} = \operatorname{expm}(-t\mathcal{L})$, $\mathcal{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$, $t > 0$ (Chung & Graham, 1997).
  - *Absorption kernel*: $K_t^{\text{Abs}} = (tA + L)^{-1}$, $t > 0$ (Jacobsen & Tien, 2018).
- Markov Matrix $P = D^{-1}A$ based Kernels and Measures.
  - *Personalized PageRank similarity measure*: $K_\alpha^{\text{PPR}} = (I - \alpha P)^{-1}$, $0 < \alpha < 1$ (Page et al., 1999).
  - *Modified Personalized PageRank*: $K_\alpha^{\text{MPPR}} = (I - \alpha P)^{-1} D^{-1} = (D - \alpha A)^{-1}$, $0 < \alpha < 1$ (Kirkland & Neumann, 2012).
  - *PageRank heat similarity measure*: $K_t^{\text{HPR}} = \operatorname{expm}(-t(I - P))$, $t > 0$ (Chung, 2007).
  - *Randomized Shortest Path distance*. Using P and the matrix of the SP distances C first get Z (Yen et al., 2008):

    $$W = P \circ \exp(-\beta C);\ Z = (I - W)^{-1} \tag{2}$$

    Then $S = (Z(C \circ W)Z) \div Z$; $\bar{C} = S - \mathbf{e}\operatorname{diag}(S)^T$, and finally, $\mathcal{D}_{\text{RSP}} = (\bar{C} + \bar{C}^T)/2$. Here $\circ$ and $\div$ are element-wise multiplication and division. Kernel version $K^{\text{RSP}}(t)$ can be obtained with equation 1.
  - *Free Energy distance*. Using Z from equation 2: $Z^h = Z\operatorname{Diag}(Z)^{-1}$; $\Phi = -1/\beta \log Z^h$; $\mathcal{D}_{\text{FE}} = (\Phi + \Phi^T)/2$ (Kivimäki et al., 2014). Kernel version $K^{\text{FE}}(t)$ can be obtained with equation 1.
- Sigmoid Commute Time kernels.

Table 1: Short names of considered kernels and other measures.

Family	Short name: Plain measure	Short name: Logged measure	Full name
Adjacency matrix based kernels	Katz	logKatz	Katz kernel
	Comm	logComm	Communicability kernel
	DF	logDF	Double Factorial similarity
Laplacian based kernels	For	logFor	Forest kernel
	Heat	logHeat	Heat kernel
	NHeat	logNHeat	Normalized Heat kernel
	Abs	logAbs	Absorption kernel
Markov matrix based kernels and measures	PPR	logPPR	Personalized PageRank similarity measure
	MPPR	logMPPR	Modified Personalized PageRank
	HPR	logHPR	PageRank heat similarity measure
	RSP	-	Randomized Shortest Path kernel
	FE	-	Free Energy kernel
Sigmoid Commute Time	SCT	-	Sigmoid Commute Time kernel
	SCCT	-	Sigmoid Corrected Commute Time kernel
	SP-CT	-	linear combination of Shortest Path and Commute Time

– *Sigmoid Commute Time kernel*:

$$K_t^{\text{SCT}} = \sigma(-tK^{\text{CT}}/\text{std}(K^{\text{CT}})),\ t > 0 \tag{3}$$

where σ is an element-wise sigmoid function $\sigma(x) = 1/(1 + e^{-x})$ (Yen et al., 2007).

– *Sigmoid Corrected Commute Time kernel*. First of all, we need Corrected Commute Time kernel (Luxburg et al., 2010):

$$K^{\text{CCT}} = HD^{-\frac{1}{2}}M(I - M)^{-1}MD^{-\frac{1}{2}}H;\ M = D^{-\frac{1}{2}}\Big(A - \frac{\vec{\mathbf{d}}\vec{\mathbf{d}}^T}{\text{vol}(G)}\Big)D^{-\frac{1}{2}}$$

where H is the centering matrix $H = I - E/n$, $\vec{\mathbf{d}}$ is the vector of diagonal elements of D and $\text{vol}(G)$ is the sum of all elements of A. Then, apply equation 3 replacing K^{CT} with K^{CCT} to obtain K^{SCCT}.

Occasionally, element-wise logarithm is applied to the resulting kernel matrix (Chebotarev, 2013; Ivashkin & Chebotarev, 2016). We apply it to almost all investigated measures and consider the resulting measures separately from their plain versions (see Table 1). For some measures, like Forest kernel, this is well-known practice (Chebotarev, 2013), while for others, like Double Factorial similarity, this transformation, to the best of our knowledge, is applied for the first time. The considered measures and their short names are summarized in Table 1.

3 Dataset

We collected a paired dataset of graphs and the corresponding results of clustering with each measure mentioned in Table 1. In this section, we describe the graph generator, the sampling strategy, the calculated graph features, and the pipeline for the measure score calculation.

We use Lancichinetti–Fortunato–Radicchi (LFR)[1] graph generator. It generates non-weighted graphs with ground truth non-overlapped communities. The model has mandatory parameters: the number of nodes n ($n > 0$), the power law exponent for the degree distribution τ_1 ($\tau_1 > 1$), the power law exponent for the community size distribution τ_2 ($\tau_2 > 1$), the fraction of intra-community edges incident to each node μ ($0 \leq \mu \leq 1$), and either minimum degree (min degree) or average

[1]NetworkX implementation: `https://networkx.github.io/documentation/networkx-2.1/reference/algorithms/generated/networkx.algorithms.community.community_generators.LFR_benchmark_graph.html`

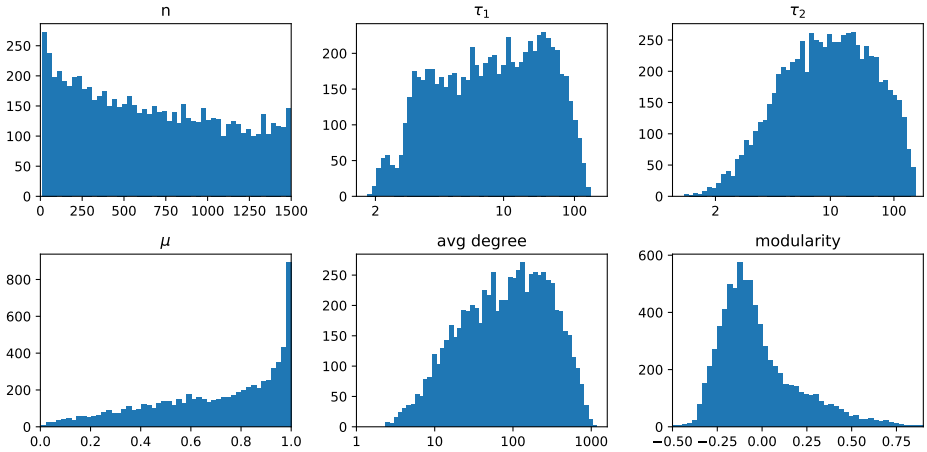


Figure 1: Distribution of graph features in the dataset.

degree (avg degree). There are also extra parameters: maximum degree (max degree), minimum community size (min community), maximum community size (max community).

For the generation, we consider $10 < n < 1500$. It is impossible to generate a dataset with a uniform distribution of all LFR parameters, because τ_1 and τ_2 parameters are located on rays. We choose to transform τ_1 and τ_2 to $\tilde{\tau}_i = 1 - (1/\sqrt{\tau_i}), i = 1, 2$ to bring their scope to $[0, 1]$ interval. Also, as avg degree feature is limited by n of particular graph, we decided to replace it with density (avg degree$/(n-1)$). It is not dependent on n and belongs to $[0, 1]$. Using all these considerations, we collected our dataset by uniformly sampling parameters for LFR generator from the set [n, $\tilde{\tau}_1$, $\tilde{\tau}_2$, μ, density] and generating graphs with these parameters. Additionally, we filter out non-connected graphs.

In total, we generate 7396 graphs. It is worth noting that the generator fails for some sets of parameters, so the resulting dataset is not uniform (see Fig. 1). In our study, non-uniformity is not a very important issue, because we are interested in local effects, not global leadership. Moreover, true uniformity for LFR parameter space is impossible, due to the unlimited scope of parameters.

For our research, we choose a minimum set of the features that describe particular properties of graphs and are not interchangeable. The LFR parameters can be divided in three groups by the graph properties they reflect:

- The size of the graph and the communities: n, τ_1, min community, max community;
- The density and uniformity of the node degrees distribution: τ_2, min degree, avg degree, max degree. As avg degree depends on n, it is distributed exponentially, so we use $\log$(avg degree) instead;
- The cluster separability: μ. As μ parameter considers only the ratio between the number of inter-cluster edges and the number of nodes but ignores overall density, we use modularity (Newman & Girvan, 2004) as a more appropriate measure for cluster separability.

Thus, the defined set of features [n, τ_1, τ_2, avg degree, modularity] is enough to consider all graph properties mentioned above.

For every generated graph we calculate the top ARI score for every measure (Hubert & Arabie, 1985). We choose ARI as a clustering score which is both popular and unbiased (Gösgens et al., 2019). As soon as every measure has a parameter, we perform clustering for a range of parameter values (we transform the parameter to be defined on [0, 1] interval and then choose 16 values linearly spaced from 0 to 1). For each value, we run 6+6+6 trials of k-means (6 trials for each of three initialization methods).

Fig. 2 shows the pipeline we use to calculate ARI score for a given LFR parameter set, a measure, and a measure parameter. Measure parameters are not the subject of our experiments, so for every

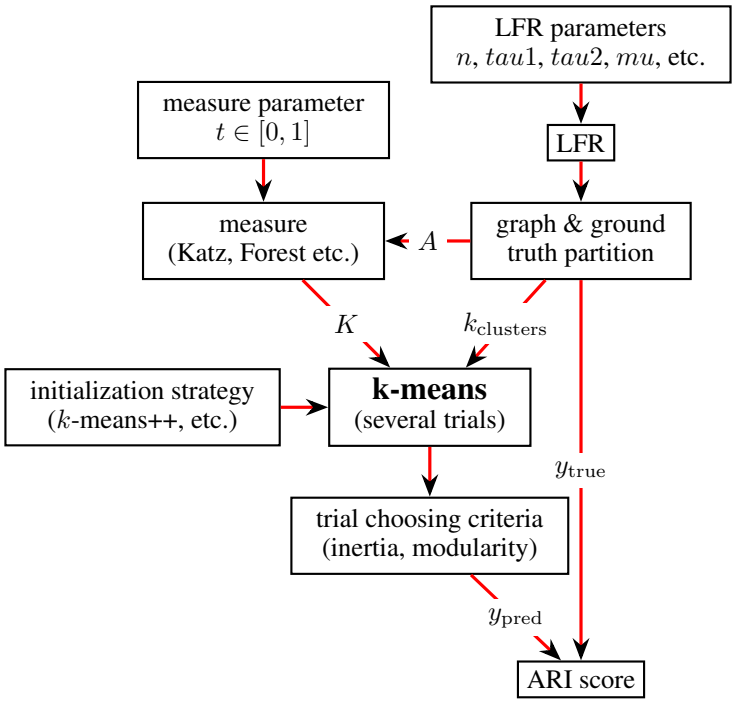


Figure 2: Measuring ARI clustering score for a particular graph, measure, and measure parameter

measure we just take the result of the measure with the value of the parameter that gives the best ARI score.

4 Results

4.1 Global leadership

We range the measures by their ARI score on every graph of the dataset. The rank is defined as the position of the measure in this list, averaged over the dataset (see Table 2a). It is important to note that the global leadership does not give a comprehensive advice on which measure is better to use, because for a particular graph, the global leader can perform worse than the others.

Some graphs have several winning measures, it can be caused by a particular graph structure or the fact that the graph is simple enough to be clustered absolutely correctly with several measures. In our dataset, 5476 out of 7396 graphs have single leader measure. Table 2b shows leaderboard only for these graphs. Assuming that measures dominate in some zones of graph parameter space, it is more important for us to consider the number of graphs where the measure wins, rather than its rank. To ensure that a measure has a dominance zone, it needs to have a large number of victories. We will check below if these victories are actually localized in some compact regions.

As SCCT is the winner in both leaderboards, we can say for sure that it is the global winner for all LFR space graphs. Other measures still can be leaders in some zones of the feature space.

4.2 Feature importance study

First of all, we find which graph features are important for the choice of the best measure and which are not. To do that, we use Linear Discriminant Analysis (Mika et al., 1999) (LDA). This method finds a new basis in feature space to classify dataset in the best way. It also shows how many components of basis are required to fit the majority of data.

Fig. 3a shows that the first two components take about 90% of the explained variance. Fig. 3b shows that these components include only τ_1, avg degree, and modularity. The fact that n is not used means

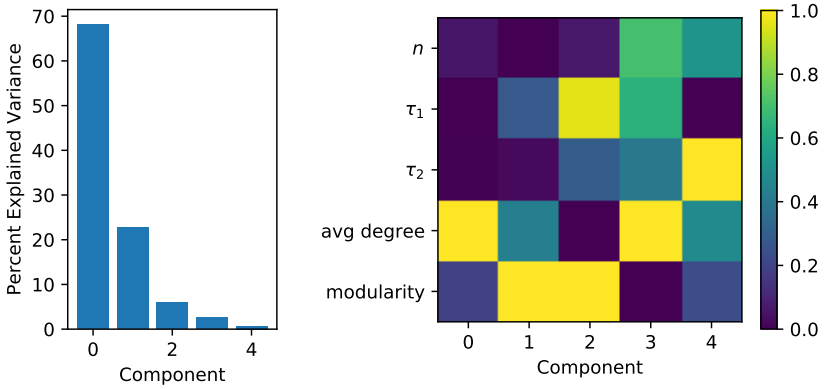
Table 2: Leaderboards

#	Measure	Rank
1	SCCT	4.7
2	NHeat	6.9
3	RSP	7.6
4	SCT	8.3
5	Comm	8.5
6	logNHeat	8.8
7	SP-CT	9.0
8	logHeatPR	9.2
9	FE	9.4
10	Katz	9.8
11	logKatz	9.9
12	logPPR	10.0
13	logModifPPR	10.5
14	DF	11.0
15	logAbs	12.1
16	logComm	12.6
17	logFor	13.0
18	HeatPR	13.4
19	logHeat	13.6
20	Heat	14.7
21	logDF	14.8
22	PPR	16.7
23	Abs	18.6
24	For	19.3
25	ModifPPR	20.5

(a) Overall rank

#	Measure	Wins	%
1	SCCT	2698	49.3
2	NHeat	837	15.3
3	Comm	268	4.9
4	logComm	244	4.5
5	RSP	168	3.1
6	SCT	139	2.5
7	logDF	128	2.3
8	SP-CT	128	2.3
9	logKatz	116	2.1
10	Abs	80	1.5
11	DF	65	1.2
12	Katz	54	1.0
13	FE	39	0.7
14	Heat	38	0.7
15	logFor	35	0.6
16	logHeat	33	0.6
17	logPPR	30	0.5
18	logModifPPR	23	0.4
19	PPR	19	0.3
20	HeatPR	19	0.3
21	logNHeat	11	0.2
22	logHeatPR	10	0.2
23	logAbs	9	0.2
24	ModifPPR	6	0.1
25	For	4	0.1

(b) Wins



(a) Explained variance. (b) Features contribution to LDA components.

Figure 3: The results of LDA components.

that the size of the graph as well as the density are not important for choosing the best measure. The fact that τ_2 is not used means that the difference in sizes of clusters is not important, too.

Fig. 4 shows the point cloud projected on the space of the two main components of LDA. We see a confirmation that the measures are indeed zoned, but the areas are quite noisy. To detect zones of measure leadership, we need to know the leadership on average in every area of space, rather than the wins in particular points. To define the local measure leadership in the whole space, we need to introduce a filtering algorithm that for every point of space returns the measure leadership

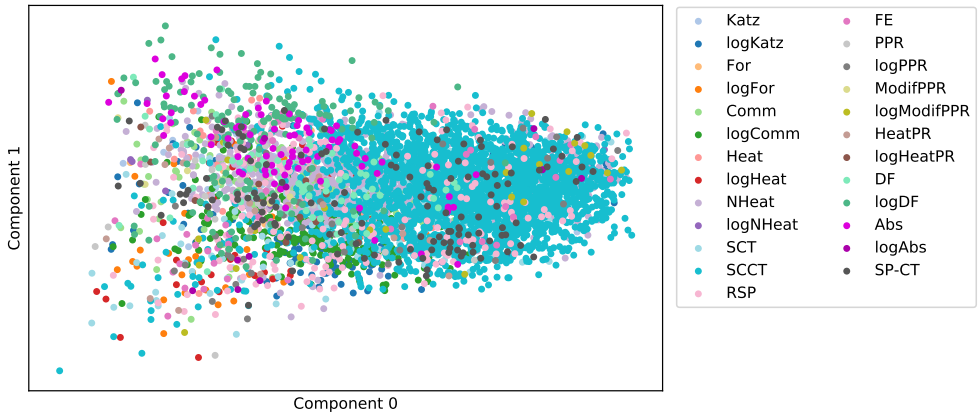
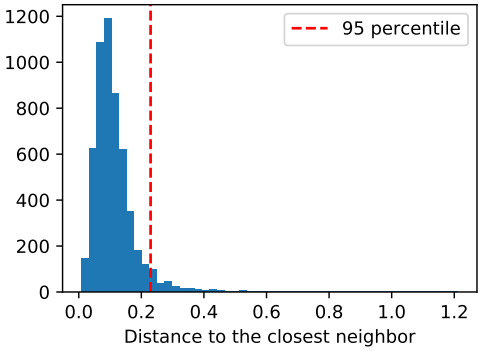
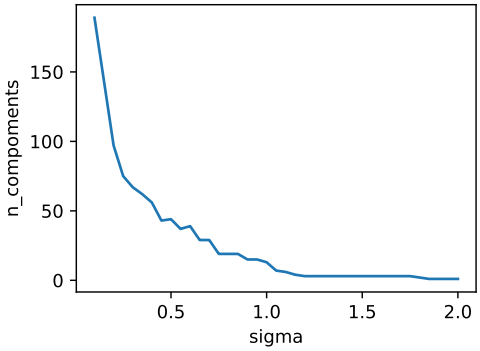


Figure 4: The dataset projected on the two main components of LDA. Point color represents the winning measure for each graph.



(a) The closest neighbor distance distribution.



(b) Number of connected components depending on σ.

Figure 5: Closest neighbor distance distribution in [τ_1, avg degree, modularity] space.

depending on the closest data points. As the choice of measure is actually dependent only on three features, we can limit our feature space to [τ_1, avg degree, modularity].

4.3 Gaussian filter in feature space

With a filter in feature space, we can suppress the noise and find actual zones of leadership for the measures. We use Gaussian filter with a scale parameter σ. For every requested point of space, it takes data points that are closer than 3σ and averages ARIs of the chosen points with a weight $e^{-\text{dist}^2/2\sigma^2}$. This allows to give larger weights to closer points. If there are less than 3 data points inside the sphere with a 3σ radius, the filter returns nothing, allowing to ignore the points with insufficient data.

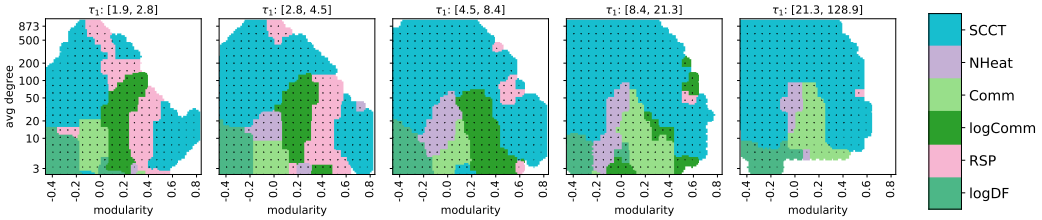
Before applying the filter, we prepare the dataset. First, we only take the points with one winning measure (see Table 2b), because multiple winners can confuse the filter. Then we normalize the standard deviation of every feature distribution to one. Finally, we cut off long tail of distant data points Fig. 5a. The resulting number of graphs is 5201.

To choose σ, we apply the filter with different sigma and look at the number of connected components in the feature space Fig. 5b. Sigma should be large enough to suppress the noise, however, it should not suppress small zones. Guided by this heuristic, we choose $\sigma = 0.5$.

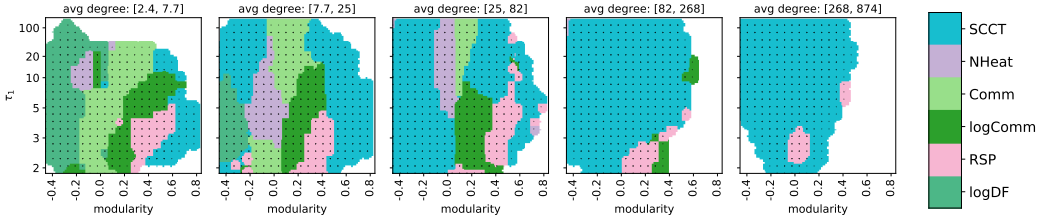
Table 3: The leaderboard of measure wins after the filtering with $\sigma = 0.5$.

Measure	SCCT	logComm	NHeat	Comm	logDF	RSP	FE	Abs	SCT	logNHeat	logFor
Wins	4283	441	268	78	64	56	3	3	2	1	1

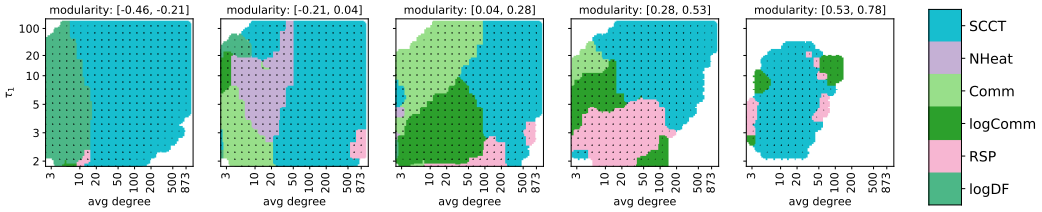
After the filtering with $\sigma = 0.5$, the leaderboard of measure wins is changed (see Table 3). Only 6 measures keep their positions: SCCT, NHeat, logComm, Comm, logDF, and RSP. This means that these measures do have zones of leadership, otherwise they would be filtered out. We can plot the entire feature space colored by the leadership zones of the measures (see Fig. 6). As the final space is 3D, we show slices of it by each of the three coordinates.



(a) Slices by τ_1.



(b) Slices by avg degree.



(c) Slices by modularity.

Figure 6: The feature space [τ_1, avg degree, modularity] divided into the leadership zones of 6 measures.

The zones of different measure leadership can be described by the following approximate criteria:

- SCCT: avg degree $>$ 100 or modularity $>$ 0.5, also modularity $<$ and avg degree $>$ 20
- NHeat: -0.20..0.20 of modularity, 0.4..0.8 τ_1, 7..55 avg degree
- logComm: 0..0.27 modularity, 0.27..0.7 τ_1, 7..55 avg degree
- Comm: -0.2..0.3 modularity, any τ_1, 0..20 avg degree
- logDF: -0.50..-0.20 of modularity, 2..20 avg degree
- RSP: 0.15..0.5 modularity, 0.27..0.5 τ_1, any avg degree.

5 CONCLUSIONS

In this work, we show that the global leadership of measures is not comprehensive. We demonstrate that among 25 measures, SCCT is the best measure for LFR graphs both by winning rate and ranking. But there are also smaller confident zones of leadership for NHeat, Comm, logComm, logDF, and RSP.

LogComm was first introduced in Ivashkin & Chebotarev (2016) and won in the competitions on graphs generated with a fixed set of Stochastic Block Model parameters. This study confirms its leadership, but only for a certain type of graphs. Another interesting finding is logDF, which unexpectedly shows good performance for the graphs with low modularity and low average degree.

This study is limited by the LFR graphs model. For other graphs (including real ones) the results can differ. The applicability of the LFR model to real life is beyond the scope of this paper.

We note that our study insensitive to non-uniformity of the dataset. While the manipulations with dataset can change the global leaderboard, they do not affect the local leadership investigated in our study.

REFERENCES

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

Konstantin Avrachenkov, Pavel Chebotarev, and Dmytro Rubanov. Kernels on graphs as proximity measures. In *International Workshop on Algorithms and Models for the Web-Graph*, pp. 27–41. Springer, 2017.

RB Bapat. Resistance distance in graphs. *Mathematics Student-India*, 68(1-4):87–98, 1999.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286 (5439):509–512, 1999.

Michael J Barber and John W Clark. Detecting network communities by propagating labels under constraints. *Physical Review E*, 80(2):026129, 2009.

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, 2008.

Ingwer Borg and Patrick JF Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

Fred Buckley and Frank Harary. *Distance in graphs*. Addison-Wesley, 1990.

Ashok K Chandra, Prabhakar Raghavan, Walter L Ruzzo, Roman Smolensky, and Prasoon Tiwari. The electrical resistance of a graph captures its commute and cover times. *Computational Complexity*, 6(4):312–340, 1996.

P Yu Chebotarev and EV Shamis. On the proximity measure for graph vertices provided by the inverse laplacian characteristic matrix. In *Fifth Conference of the International Linear Algebra Society*, pp. 30–31, 1995.

P Yu Chebotarev and EV Shamis. On a duality between metrics and σ-proximities. *Automation and Remote Control*, 1998a.

Pavel Chebotarev. Studying new classes of graph metrics. In *International Conference on Geometric Science of Information*, pp. 207–214. Springer, 2013.

Pavel Chebotarev and Elena Shamis. On proximity measures for graph vertices. *Automation and Remote Control*, 1998b.

Fan Chung. The heat kernel as the pagerank of a graph. *Proceedings of the National Academy of Sciences*, 104(50):19735–19740, 2007.

Fan Chung and Shing-Tung Yau. Coverings, heat kernels and spanning trees. *Journal of Combinatorics*, 6:163–184, 1998.

Fan RK Chung and Fan Chung Graham. *Spectral Graph Theory*. Number 92 in 1. American Mathematical Soc., 1997.

Peter G Doyle and J Laurie Snell. *Random Walks and Electric Networks*, volume 22. American Mathematical Soc., 1984.

Anton J Enright, Stijn Van Dongen, and Christos A Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleic acids research*, 30(7):1575–1584, 2002.

Ernesto Estrada and Naomichi Hatano. Statistical-mechanical approach to subgraph centrality in complex networks. *Chemical Physics Letters*, 439(1-3):247–251, 2007.

Ernesto Estrada and Naomichi Hatano. Communicability in complex networks. *Physical Review E*, 77(3):036111, 2008.

Ernesto Estrada and Grant Silver. Accounting for the role of long walks on networks via a new matrix function. *Journal of Mathematical Analysis and Applications*, 449(2):1581–1600, 2017.

Francois Fouss, Luh Yen, Alain Pirotte, and Marco Saerens. An experimental investigation of graph kernels on a collaborative recommendation task. In *Sixth International Conference on Data Mining (ICDM'06)*, pp. 863–868. IEEE, 2006.

François Fouss, Marco Saerens, and Masashi Shimbo. *Algorithms and models for network data and link analysis*. Cambridge University Press, 2016.

F Göbel and AA Jagers. Random walks on graphs. *Stochastic processes and their applications*, 2 (4):311–336, 1974.

Martijn Gösgens, Liudmila Prokhorenkova, and Alexey Tikhonov. Systematic analysis of cluster similarity indices: Towards bias-free cluster validation. *arXiv preprint arXiv:1911.04773*, 2019.

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

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

Vladimir Ivashkin and Pavel Chebotarev. Do logarithmic proximity measures outperform plain ones in graph clustering? In *International Conference on Network Analysis*, pp. 87–105. Springer, 2016.

Karly A Jacobsen and Joseph H Tien. A generalized inverse for graphs with absorption. *Linear Algebra and its Applications*, 537:118–147, 2018.

Jaz Kandola, Nello Cristianini, and John S Shawe-Taylor. Learning semantic similarity. In *Advances in neural information processing systems*, pp. 673–680, 2003.

Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

Stephen J Kirkland and Michael Neumann. *Group inverses of M-matrices and their applications*. CRC Press, 2012.

Ilkka Kivimäki, Masashi Shimbo, and Marco Saerens. Developments in the theory of randomized shortest paths with a comparison of graph node distances. *Physica A: Statistical Mechanics and its Applications*, 393:600–616, 2014.

Douglas J Klein and Milan Randić. Resistance distance. *Journal of mathematical chemistry*, 12(1): 81–95, 1993.

Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4):046110, 2008.

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

Ulrike V Luxburg, Agnes Radl, and Matthias Hein. Getting lost in space: Large sample analysis of the resistance distance. In *Advances in Neural Information Processing Systems*, pp. 2622–2630, 2010.

James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, number 14 in 1, pp. 281–297. Oakland, CA, USA, 1967.

Sebastian Mika, Gunnar Ratsch, Jason Weston, Bernhard Scholkopf, and Klaus-Robert Mullers. Fisher discriminant analysis with kernels. In *Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop (cat. no. 98th8468)*, pp. 41–48. Ieee, 1999.

DJH Moore and GE Subak-Sharpe. Metric transformation of an (m+ 1)-terminal resistive network into a hyperacute angled simplex pm in euclidean space em. In *Proceedings of the Eleventh Midwest Symposium on Circuit Theory, Notre Dame, Indiana*, pp. 184–192, 1968.

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

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

Muhammad Qasim Pasta and Faraz Zaidi. Topology of complex networks and performance limitations of community detection algorithms. *IEEE Access*, 5:10901–10914, 2017.

Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3):036106, 2007.

GE Sharpe. Solution of the (m+ 1)-terminal resistive network problem by means of metric geometry. In *Proceedings of the First Asilomar Conference on Circuits and Systems, Pacific Grove, CA*, pp. 319–328, 1967.

John Shawe-Taylor, Nello Cristianini, et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

Felix Sommer, François Fouss, and Marco Saerens. Comparison of graph node distances on clustering tasks. In *International Conference on Artificial Neural Networks*, pp. 192–201. Springer, 2016.

Felix Sommer, François Fouss, and Marco Saerens. Modularity-driven kernel k-means for community detection. In *International Conference on Artificial Neural Networks*, pp. 423–433. Springer, 2017.

Stijn Marinus Van Dongen. *Graph clustering by flow simulation*. PhD thesis, 2000.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Luh Yen, Francois Fouss, Christine Decaestecker, Pascal Francq, and Marco Saerens. Graph nodes clustering based on the commute-time kernel. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 1037–1045. Springer, 2007.

Luh Yen, Marco Saerens, Amin Mantrach, and Masashi Shimbo. A family of dissimilarity measures between nodes generalizing both the shortest-path and the commute-time distances. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 785–793, 2008.

Luh Yen, Francois Fouss, Christine Decaestecker, Pascal Francq, and Marco Saerens. Graph nodes clustering with the sigmoid commute-time kernel: A comparative study. *Data & Knowledge Engineering*, 68(3):338–361, 2009.