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 graph generator, 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 takes into account power law distribution of node degrees and community sizes. There are other generation models, e.g., Naive Scale-free Clustering (Pasta & Zaidi, 2017). We choose the LFR model: although it misses key properties of real graphs, like diameter or the clustering coefficient, this model has been proven to be effective in meta-learning (Prokhorenkova, 2019).

There are a lot of measure benchmarking studies considering node classification and clustering for both generated graphs and datasets (Fouss et al., 2012; Sommer et al., 2016; 2017; Avrachenkov et al., 2017; Ivashkin & Chebotarev, 2016; Guex et al., 2018; 2019; Aynulin, 2019a;b; Courtain et al., 2020; Leleux et al., 2020), etc. Although a large number of experimental results, theoretical results still look unattainable. One of the most important theoretical results for graph measures is a work Luxburg et al. (2010), where problems of Commute Time on big graphs were shown theoretically, and a substantiated amendment was proposed to correct the problem. The paper shows how difficult such proves. Besides difficult proves, there is still no complex empirical understanding of what effects need to be proven. Our empirical work has two main advantages from previous ones. Firstly, we consider the vast amount of graph measures, which for the first time gives the full picture. Secondly, 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;
- We check applicability of the results on real-world graphs.

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, k-means++ (Arthur & Vassilvitskii, 2006) and random partition. We combine all these strategies to reduce the impact of the initialization strategy on the result.

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 $\kappa : V(G) \times V(G) \to \mathbb{R}$, 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. Possibility of the approach was mentioned in Fouss et al.

(2016). 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. 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* $K_t^{\text{Comm}} = \exp\!\text{m}(tA)$, $t > 0$. 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. $D = \text{Diag}(A \cdot \mathbf{1})$ is the degree matrix of G, $\text{Diag}(\mathbf{x})$ 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*: $K_t^{\text{Heat}} = \exp\!\text{m}(-tL)$, $t > 0$ (Chung & Yau, 1998).
  - *Normalized Heat kernel*: $K_t^{\text{NHeat}} = \exp\!\text{m}(-t\mathcal{L})$, $\mathcal{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$, $t > 0$ (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*: $K_t^{\text{HPR}} = \exp\!\text{m}(-t(I - P))$, $t > 0$ (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 $S = (Z(C \circ W)Z) \div Z$; $\bar{C} = S - \mathbf{e}\,\text{diag}(S)^T$, 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: $Z^h = Z\,\text{Diag}(Z)^{-1}$; $\Phi = -1/\beta \log Z^h$; $\mathcal{D}_{\text{FE}} = (\Phi + \Phi^T)/2$ (Kivimäki et al., 2014). Kernel version $K^{\text{FE}}(t)$ can be obtained with equation 1.
- Sigmoid Commute Time kernels.
  - *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).

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 SP and CT

– *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) 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 degree (avg degree). There are also extra parameters: maximum degree (max degree), minimum community size (min community), maximum community size (max community). Not all LFR parameters space corresponds real world graphs, usually real graphs correspond to $\tau_1 \in [1, 4]$ and $\mu < 0.5$. However, there is also separated interesting case of bisect graphs ($\mu > 0.5$). We still consider the entire parameter space to cover all the cases.

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. In this case, realistic setting $\tau_1 \in [1, 4]$ has 50% of all variable range. Also, as avg degree feature is limited by n

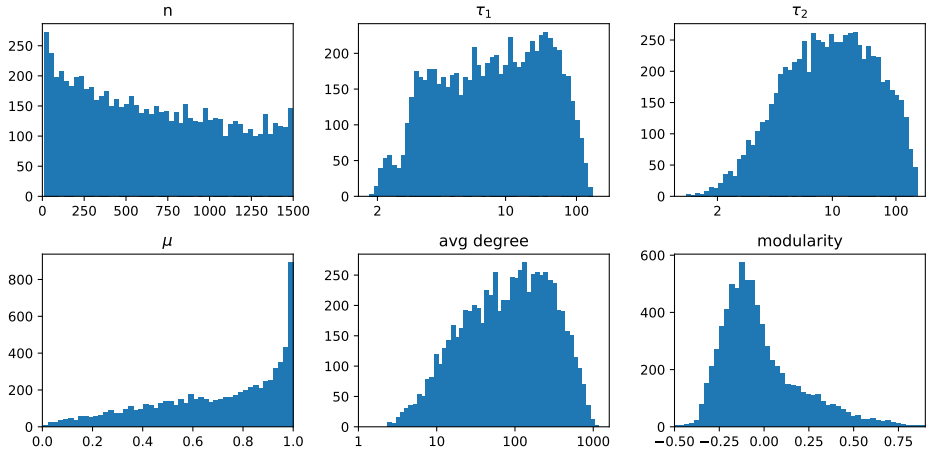


Figure 1: Distribution of graph features in the dataset.

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.

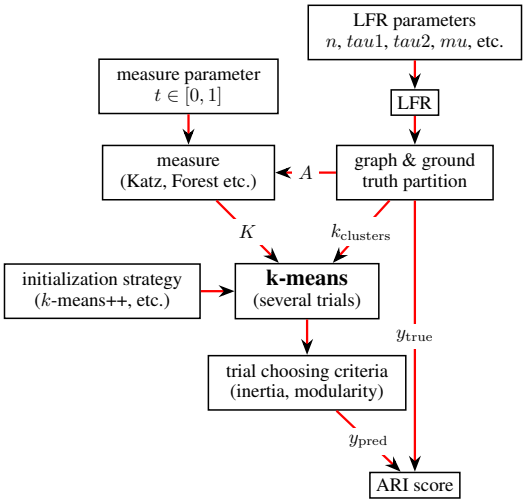


Figure 2: Measuring ARI clustering score for a particular graph, measure, and measure parameter

Thus, the defined set of features [n, τ_1, τ_2, avg degree, modularity] is enough to consider all graph properties mentioned above. Although modularity is widely used measure, it suffers from resolution limit problems (Fortunato & Barthelemy, 2007). We acknowledge that the use of this measure may negatively affect the quality of our method.

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.,

Table 2: Overall leaderboard. Percentage of wins is calculated among all 7396 graphs of the dataset. Column ARI stands for mean ARI across the dataset.

#	Measure	Rank	Wins, %	ARI
1	SCCT	4.7	56.0	0.58
2	NHeat	6.9	25.2	0.40
3	RSP	7.7	22.4	0.42
4	SCT	8.4	19.5	0.40
5	Comm	8.4	15.8	0.36
6	logNHeat	8.8	18.0	0.37
7	SP-CT	9.0	20.4	0.41
8	logHeatPR	9.2	18.0	0.37
9	FE	9.5	19.3	0.39
10	Katz	9.8	7.7	0.32
11	logKatz	9.9	18.1	0.35
12	logPPR	10.0	17.5	0.35
13	logModifPPR	10.5	17.3	0.35
14	DF	11.0	8.1	0.28
15	logAbs	12.1	15.9	0.33
16	logComm	12.6	15.5	0.23
17	logFor	13.0	12.7	0.22
18	HeatPR	13.4	12.5	0.22
19	logHeat	13.6	11.9	0.21
20	Heat	14.7	11.2	0.19
21	logDF	14.8	11.1	0.18
22	PPR	16.7	4.0	0.13
23	Abs	18.6	3.7	0.08
24	For	19.3	2.8	0.07
25	ModifPPR	20.5	1.7	0.05

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

Due to the fact that in the task it is necessary to iterate over the graphs, measures, parameter values, initializations—the task is computationally difficult. Overall computation time is 20 days on 18 CPU cores and 6 GPUs.

4 RESULTS

4.1 GLOBAL LEADERSHIP IN LFR SPACE

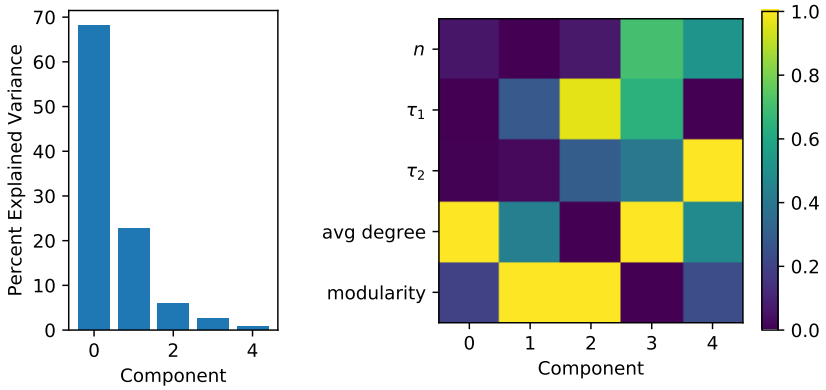
We range the measures by their ARI score on every graph of the dataset. The rank is defined as the position of the measure in this list, averaged over the dataset (see Table 2). It is important to note that the global leadership does not give a comprehensive advice on which measure is better to use, because for a particular graph, the global leader can perform worse than the others. Here we consider the entire LFR space, not just its zone corresponding to real graphs, so the ranking may differ from similar works.

As SCCT is the winner for both ranking and percentage of wins, 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 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.



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

Figure 3: The results of LDA components.

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 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].

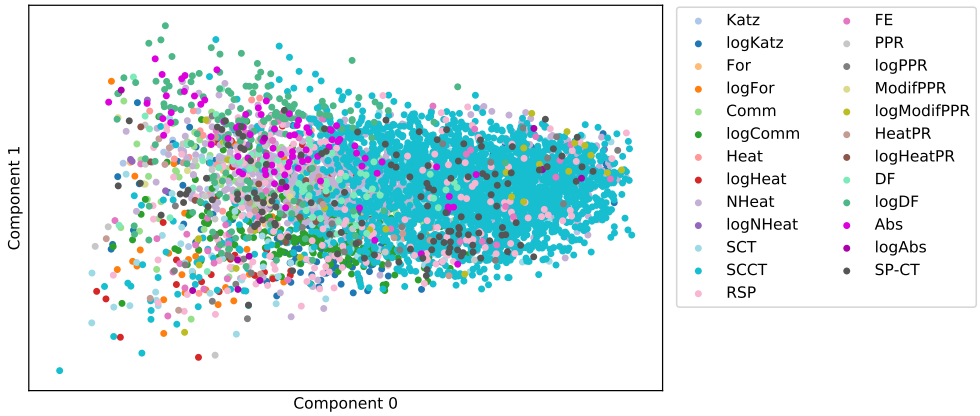


Figure 4: The dataset projected on the two main components of LDA. Point color represents the winning measure for each graph.

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 only one winning measure, because multiple winners can confuse the filter. Then we normalize the standard

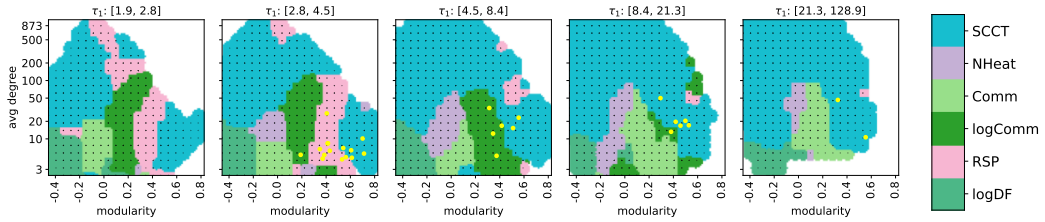
deviation of every feature distribution to one. Finally, we cut off long tail of distant data points The resulting number of graphs is 5201.

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

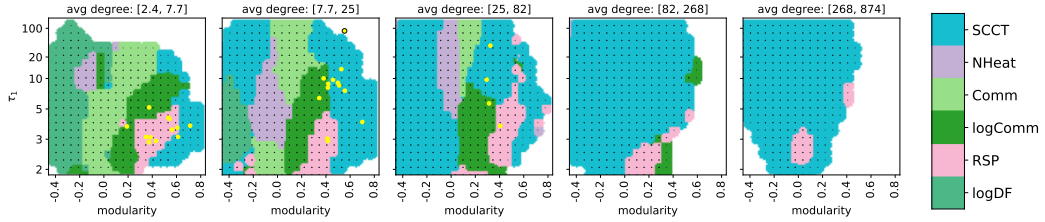
Table 3: The leaderboard of measure wins after the filtering with $\sigma = 0.5$.

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

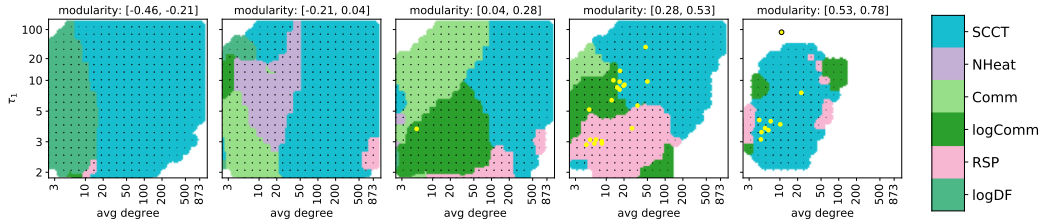
After the filtering with $\sigma = 0.5$, the leaderboard of measure wins is changed (see Table 3). Only 6 measures keep their positions: SCCT, NHeat, logComm, Comm, logDF, and RSP. This means that these measures do have zones of leadership, otherwise they would be filtered out. We can plot the entire feature space colored by the leadership zones of the measures (see Fig. 5). 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 5: The feature space [τ_1, avg degree, modularity] divided into the leadership zones of 6 measures. Yellow points represent position of real-world datasets in the space.

5 DATASETS

Even though LFR lacks some characteristics of real-world graphs, there is evidence that the optimal parameter of the Louvain clustering for a real graph is close to the parameter for LFR graphs gen-

erated from the parameters of a real one (Prokhorenkova, 2019). So, there is a chance that learned space might be helpful for choosing measures in the wild.

For evaluation, we use 29 graphs of standard datasets: Dolphins (Lusseau et al., 2003), Football (Newman & Girvan, 2004), Karate club (Zachary, 1977), Newsgroup (9 subsets, weights are binarized with threshold 0.1) (Yen et al., 2007), Political blogs (Adamic & Glance, 2005), Political books (Newman, 2006), SocioPatterns Primary school day (2 graphs) (Stehlé et al., 2011), Cora (11 subsets) (McCallum et al., 2000), Eu-core (Leskovec et al., 2007), EuroSIS (WebAtlas, 2009). Parameters of graphs are marked on Fig. 5. For each graph, we found the best ARI for every measure (iterating over the measure parameter value). Now we can check the quality of measure choice, based on calculated LFR data. Result of recommendation is the measure which will be chosen for particular set of parameters.

Table 4: Mean ARI of LFR recommendations strategies for datasets. Top6 stands for the set of measures which have zones in LFR parameter space.

Strategy	Mean ARI
Always SCCT	0.615
Based on LFR space, top6 measures	**0.620**
Based on LFR space, all measures	0.619
Upper bound	0.643

The best measures on datasets are SCCT (by mean ARI) and SCT (by rank). It is close to the results obtained for LFR. Moreover, the correlation between ranks of measures for datasets and for corresponding LFR recommendations is 0.90.

Let us use "always use SCCT" as our baseline strategy. In Table 4 we compare it with strategies based on LFR space. We obtain LFR recommendation using knn as a well-proven method for meta-learning. Since each graph is unique, the result of 1nn can be very noisy, so that we will use 5nn.

Table 4 shows that recommendation approach slightly beat the baseline. However, it is not enough to draw confident conclusions about the applicability of the method. Using this fact and the fact that the ranks of datasets and recommendations are highly correlated, we conclude that building a meta-learning procedure is adequate to give a robust recommendation, but not precise enough to beat baseline confidently. It could be connected with the fact that node labels of real graphs do not obey a certain system since they were created in the wild. A larger dataset could help separate the signal from the noise and pinpoint where the limits of the method are. At least, the good news is that the conclusions made on the LFR do not contradict the results that can be obtained on the datasets.

6 Conclusions

In this work, we show that the global leadership of measures doesn't give comprehension knowledge about graph measures. 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.

Our work do not contradict with other experimental papers, but expands them and gives new findings. 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 based on LFR benchmark data. An attempt to apply the results to real data gives quality at the baseline level. However, there is a strong correlation between the ranking of measures for datasets and the ranking of LFR recommendation, which indicates that that the leadership measures are the same, but the recommendations are not accurate.

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

Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pp. 36–43, 2005.

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.

Rinat Aynulin. Efficiency of transformations of proximity measures for graph clustering. In *International Workshop on Algorithms and Models for the Web-Graph*, pp. 16–29. Springer, 2019a.

Rinat Aynulin. Impact of network topology on efficiency of proximity measures for community detection. In *International Conference on Complex Networks and Their Applications*, pp. 188–197. Springer, 2019b.

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.

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 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.

Pavel Chebotarev and Elena Shamis. On a duality between metrics and σ-proximities. *Automation and Remote Control*, 1998a.

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.

Sylvain Courtain, Pierre Leleux, Ilkka Kivimäki, Guillaume Guex, and Marco Saerens. Randomized shortest paths with net flows and capacity constraints. *Information Sciences*, 2020.

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.

Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proceedings of the national academy of sciences*, 104(1):36–41, 2007.

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, Kevin Francoisse, Luh Yen, Alain Pirotte, and Marco Saerens. An experimental investigation of kernels on graphs for collaborative recommendation and semisupervised classification. *Neural networks*, 31:53–72, 2012.

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.

Guillaume Guex, Ilkka Kivimäki, and Marco Saerens. Randomized optimal transport on a graph: framework and new distance measures. *arXiv preprint arXiv:1806.03232*, 2018.

Guillaume Guex, Sylvain Courtain, and Marco Saerens. Covariance and correlation kernels on a graph in the generalized bag-of-paths formalism. *arXiv preprint arXiv:1902.03002*, 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.

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

Pierre Leleux, Sylvain Courtain, Guillaume Guex, and Marco Saerens. Sparse randomized shortest paths routing with tsallis divergence regularization. *arXiv preprint arXiv:2007.00419*, 2020.

Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2–es, 2007.

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

David Lusseau, Karsten Schneider, Oliver J Boisseau, Patti Haase, Elisabeth Slooten, and Steve M Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.

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.

Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

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.

Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

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.

Liudmila Prokhorenkova. Using synthetic networks for parameter tuning in community detection. In *International Workshop on Algorithms and Models for the Web-Graph*, pp. 1–15. Springer, 2019.

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.

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.

Juliette Stehlé, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Lorenzo Isella, Jean-François Pinton, Marco Quaggiotto, Wouter Van den Broeck, Corinne Régis, Bruno Lina, et al. High-resolution measurements of face-to-face contact patterns in a primary school. *PloS one*, 6(8):e23176, 2011.

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.

WebAtlas. Eurosis webmapping, 2009. URL `http://www.webatlas.fr/exhibition/eurosis/`.

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.

Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977.