Classification Supported by Community-Aware Node Features

Bogumił Kamiński[1], Paweł Prałat[2], François Théberge[3], and Sebastian Zając[1]

[1] Decision Analysis and Support Unit, SGH Warsaw School of Economics, Warsaw, Poland; e-mail: {bkamins,szajac2}@sgh.waw.pl
[2] Department of Mathematics, Toronto Metropolitan University, Toronto, ON, Canada; e-mail: pralat@torontomu.ca
[3] Tutte Institute for Mathematics and Computing, Ottawa, ON, Canada; email: theberge@ieee.org

Abstract. A community structure that is often present in complex networks plays an important role not only in their formation but also shapes dynamics of these networks, affecting properties of their nodes. In this paper, we propose a family of community-aware node features and then investigate their properties. We show that they have high predictive power for classification tasks. We also verify that they contain information that cannot be recovered neither by classical node features nor by classical or structural node embeddings.

Keywords: node classification, community structure, node features

1 Introduction

In the context of relational data, node classification is a particularly important problem in which data is represented as a network and the goal is to predict labels associated with its nodes. It is widely used in various practical applications such as recommender systems [27], social network analysis [3], and applied chemistry [12].

However, for classifiers to perform well, they must have access to a set of highly informative node features that can discriminate representatives of different classes. No matter how sophisticated classifiers one builds, they will perform poorly as long as they do not get informative input concerning the problem at hand. Hence, it is desirable to enrich a family of available features and apply machine learning tools to features of various sorts.

In this paper, we investigate a family of features that pay attention to community structure that is often present in complex networks and plays an important role in their formation, affecting nodes' properties. Such features are further called *community-aware features*. Indeed, community structure of real-world networks often reveals the internal organization of nodes [10]. Such communities form groups of densely connected nodes with substantially less edges touching other parts of the graph. Identifying communities in a network can be done in an unsupervised way and is often the first step the analysts take.

The motivation to study community-aware features is twofold. On one hand, one can expect that for many node classification tasks such features can be highly informative. For example, it might be important whether a given node is a strong community member or, conversely, it is loosely tied to many communities. On the other hand, one can expect that community-aware features are not highly correlated to other features that are typically computed for networks. Indeed, to compute community-aware features one needs first to identify the community structure of a graph. This, in turn, is a complicated non-linear transformation of the input graph, which cannot be expected to be easily recovered by supervised or unsupervised machine learning models that are not designed to be community-aware.

In this paper, we show that there are classes of node prediction problems in which community-aware features have high predictive power. We also verify that community-aware features contain information that cannot be recovered neither by classical node features nor by node embeddings (both classical as well as structural). In our experiments, we concentrate on binary classification to assure that the results can be reported consistently across different graphs.

There are some community-aware features already introduced in the literature such as CADA [15] or the participation coefficient [14]; see Section 2 for their definitions. However, it is important to highlight that both CADA and the participation coefficient ignore the distribution of community sizes. We argue that taking community sizes into account when computing community-aware features matters as it provides a more detailed picture. Therefore, in this paper we propose a class of community-aware features that, via the appropriate null model, take into account community sizes and compare their predictive performance to the measures that have been previously proposed in the literature.

This is a short proceeding version of a longer journal paper[4]. The longer version includes a discussion about using the null models to design ML tools, including the modularity function that is used by many clustering algorithms. More importantly, in the journal version we show how the null model is used to define one of our community-aware features, namely, the Community Association Strength (see Subsection 2.3). Due to space limitations, we also do not include in this short version various additional experiments (for example, on synthetic networks) and discussion on computational complexity of the algorithms computing node features.

2 Community-Aware Node Features

In this section, we introduce various community-aware node features. All of them aim to capture and quantify how given nodes are attached to communities. It will be assumed that a partition $\mathbf{A} = \{A_1, A_2, \ldots, A_\ell\}$ of the set of nodes V into ℓ communities is already provided; communities induced by parts A_i ($i \in [\ell]$) are

[4] The preprint of the journal version can be found on-line: https://math.torontomu.ca/~pralat/research.html

denser comparing to the global density of the graph. Such partition can be found by any clustering algorithm. In our empirical experiments we use Leiden [25] which is known to produce good, stable results.

To simplify the notation, we will use $\deg_{A_i}(v)$ to be the number of neighbours of v in A_i.

2.1 Anomaly Score CADA

The first community-aware node feature is the anomaly score introduced in [16]. The *anomaly score* is computed as follows: for any node $v \in V$ with $\deg(v) \geq 1$,

$$\mathrm{cd}(v) = \frac{\deg(v)}{d_{\mathbf{A}}(v)}, \qquad \text{where} \qquad d_{\mathbf{A}}(v) = \max\Big\{ \deg_{A_i}(v) : A_i \in \mathbf{A} \Big\};$$

the denominator, $d_{\mathbf{A}}(v)$, represents the maximum number of neighbouring nodes that belong to the same community. In one extreme, if all neighbours of v belong to the same community, then $\mathrm{cd}(v) = 1$. In the other extreme, if no two neighbours of v belong to the same community, then $\mathrm{cd}(v) = \deg(v)$.

Note that $\mathrm{cd}(v)$ does not pay attention to which community node v belongs to. Moreover, this node feature is unbounded, that is, $\mathrm{cd}(v)$ may get arbitrarily large. As a result, we will also investigate the following small modification of the original score, the *normalized anomaly score*: for any node $v \in A_i$ with $\deg(v) \geq 1$,

$$\overline{\mathrm{cd}}(v) = \frac{\deg_{A_i}(v)}{\deg(v)}.$$

Clearly, $0 \leq \overline{\mathrm{cd}}(v) \leq 1$. Moreover, any reasonable clustering algorithm typically should try to assign v to the community where most of its neighbours are, so most nodes are expected to have $\overline{\mathrm{cd}}(v) = 1/\mathrm{cd}(v)$.

2.2 Normalized Within-module Degree and Participation Coefficient

In [14], an interesting and powerful approach was proposed to quantify the role played by each node within a network that exhibits community structure. Seven different universal roles were heuristically identified, each defined by a different region in the $(z(v), p(v))$ 2-dimensional parameter space, where $z(v)$ is the normalized within-module degree of a node v and $p(v)$ is the participation coefficient of v. Node feature $z(v)$ captures how strongly a particular node is connected to other nodes within its own community, completely ignoring edges between communities. On the other hand, node feature $p(v)$ captures how neighbours of v are distributed between all parts of the partition $\mathbf{A}$.

Formally, the *normalized within-module degree* of a node v is defined as follows: for any node $v \in A_i$,

$$z(v) = \frac{\deg_{A_i}(v) - \mu(v)}{\sigma(v)},$$

where $\mu(v)$ and $\sigma(v)$ are, respectively, the mean and the standard deviation of $\deg_{A_i}(u)$ over all nodes u in the part v belongs to. If node v is tightly connected to other nodes within the community, then $z(v)$ is large and positive. On the other hand, $|z(v)|$ is large and $z(v)$ is negative when v is loosely connected to other peers.

The *participation coefficient* of a node v is defined as follows: for any node $v \in V$ with $\deg(v) \geq 1$,

$$p(v) = 1 - \sum_{i=1}^{\ell} \left(\frac{\deg_{A_i}(v)}{\deg(v)} \right)^2 .$$

The participation coefficient $p(v)$ is equal to zero if v has neighbours exclusively in one part (most likely in its own community). In the other extreme situation, the neighbours of v are homogeneously distributed among all parts and so $p(v)$ is close to the trivial upper bound of $1 - 1/\ell \approx 1$.

2.3 Community Association Strength

As already advertised, let us now introduce our own community-aware node feature that takes the distribution of community sizes into account. Its derivation is explained in the journal version of this paper. For any $v \in A_i$, we define the *community association strength* as follows:

$$\beta^*(v) = 2 \left(\frac{\deg_{A_i}(v)}{\deg(v)} - \lambda \frac{\mathrm{vol}(A_i) - \deg(v)}{\mathrm{vol}(V)} \right) .$$

The lower the value of $\beta^*(v)$, the less associated node v with its own community is. In the derivation above we allow for any $\lambda > 0$, but in the experiments, we will use $\lambda = 1$.

Let us also notice that when $\lambda = 1$, $\beta^*(v)$ is essentially twice the normalized anomaly score $\overline{\mathrm{cd}}(v)$ after adjusting it to take into account the corresponding prediction from the null model. Moreover, let us note that some simplified version of this node feature was already used in [18].

2.4 Distribution-Based Measures

Our next community-aware node features are similar in spirit to the participation coefficient, that is, they aim to measure how neighbours of a node v are distributed between all parts of the partition $\mathbf{A}$. The main difference is that they pay attention to the sizes of parts of $\mathbf{A}$ and compare the distribution of neighbours to the corresponding predictions from the null model. They are upgraded versions of the participation coefficient, similarly to the community association strength being an upgraded counterpart of the normalized anomaly score.

Formally, for any node $v \in V$, let $q_1(v)$ be the vector representing fractions of neighbours of v in various parts of partition $\mathbf{A}$. Similarly, let $\hat{q}_1(v)$ be the corresponding prediction for the same vector based on the Chung-Lu model.

Note that $\hat{q}_1(v) = \hat{q}_1$ does *not* depend on v (of course, it should not!) but only on the distribution of community sizes. Our goal is to measure how similar the two vectors are. A natural choice would be any of the p-norms but, since both vectors are stochastic (that is, all entries are non-negative and they add up to one), alternatively one can also use any good measure for comparison of probability distributions. In our experiments we tested the following node features: L^1 *norm* $L_1^1(v)$, L^2 *norm* $L_1^2(v)$, *Kullback–Leibler divergence* $\mathrm{kl}_1(v)$, and *Hellinger distance* $h_1(v)$.

The above measures pay attention to which communities neighbours of v belong to. However, some of such neighbours might be strong members of their own communities but some of them might not be. Should we pay attention that? Is having a few strong members of community A_i as neighbours equivalent to having many neighbours that are weak members of A_i? To capture these nuances, one needs to consider larger ego-nets around v, nodes at distance at most 2 from v. We define $q_2(v)$ to be the average value of $q_1(u)$ taken over all neighbours of v. As before, $\hat{q}_2(v)$ is the corresponding prediction based on the null model. However, since $\hat{q}_1(u) = \hat{q}_1$ does not depend on u, $\hat{q}_2(v)$ also does not depend on v and, in fact, it is equal to $\hat{q}_1$. The difference between $q_2(v)$ and $\hat{q}_2(v)$ may be measured by any metric used before. In our experiments we tested $L_2^1(v)$, $L_2^2(v)$, $\mathrm{kl}_2(v)$, and $h_2(v)$, counterparts of $L_1^1(v)$, $L_1^2(v)$, $\mathrm{kl}_1(v)$, and $h_1(v)$ respectively.

Let us mention that $q_1(v)$ and $q_2(v)$ have a natural and useful interpretation. Consider a random walk that starts at a given node v. The ith entry of the $q_1(v)$ vector is the probability that a random walk visits a node from community A_i after one step. Vector $q_2(v)$ has the same interpretation but after two steps are taken by the random walk.

One can repeat the same argument and define $L_i^1(v)$, etc., for any natural number i by performing i steps of a random walk. Moreover, a natural alternative approach would be to consider all possible walk lengths but connections made with distant neighbours are penalized by an attenuation factor α as it is done in the classical Katz centrality [19].

Finally, let us note that the above aggregation processes could be viewed as simplified versions of GNNs classifiers. Therefore, the investigation of these measures additionally shows how useful community-aware measures could be when used in combination with GNN models.

3 Experiments

3.1 Graphs Used

We consider undirected, connected, and simple graphs so that all node features are well defined and all methods that we use work properly. In each graph, we have some "ground-truth" labels for the nodes which are used to benchmark classification algorithms. For consistency of the reported metrics, we consider binary classification tasks, so the ground-truth node features that are to be predicted will always consist of labels from the set $\{0, 1\}$ with label 1 being the target class.

In the experiments, we used two families of graphs. The first family consists of synthetic networks, the **A**rtificial **B**enchmark for **C**ommunity **D**etection with **O**utliers (**ABCD+o**) [18]. The main goal of experiments on this family is to perform a sanity test to evaluate whether the basic functionality of community-aware node features is working correctly or not. The results that are available in the journal version of this paper show that for this class of graphs community-aware node features significantly outperform other features.

The second family of networks we used in our experiments are empirical real-world graphs. We tried to select a collection of graphs with different properties (density, community structure, degree distribution, clustering coefficient, etc.). More importantly, some of them have highly unbalanced binary classes. Experiments with these networks will serve as a more challenging and robust test for usefulness of the proposed community-aware node features.

Empirical Graphs For experiments on real-world, empirical networks, we selected the following 5 datasets. In cases when multiple connected components were present, we kept only the giant component. Self-loops, if present, were also dropped before performing the experiments. We summarize some statistics for the above graphs in Table 1.

- **Reddit** [20]: A bipartite graph with 9,998 nodes representing users in one part and 982 nodes representing subreddits in the other one. The target class represents banned users.
- **Grid** [21]: A power grid network with attributes nodes. The target class corresponds to nodes with "plant" attribute.
- **Facebook** [23]: Nodes correspond to official Facebook pages that belong to one of the 4 categories and edges are mutual likes. The target class corresponds to "politician" category.
- **LastFM** [24]: Nodes are users of the social network and edges represent mutual followers. There are some nodes attributes including the location; the target class corresponds to nodes with "country 17" attribute.
- **Amazon** [8]: Nodes are users and edges represent common product reviews. The target class corresponds to users with less than 20% "helpful" votes, and non-target correspond to users with more than 80% "helpful" votes.

Table 1. Statistics of the selected real-world empirical graphs.

dataset	# of nodes	average degree	# of clusters	target class proportion
Reddit	10,980	14.30	12	3.661%
Grid	13,478	2.51	78	0.861%
LastFM	7,624	7.29	28	20.619%
Facebook	22,470	15.20	58	25.670%
Amazon	9,314	37.49	39	8.601%

3.2 Node Features Investigated

The community-aware node features that we tested are summarized in Table 2. The features are computed with reference to a partition of a graph into communities obtained using the Leiden algorithm. The partition is chosen as the best of 1,000 independent runs of the `community_leiden` function implemented in the *igraph* library [7] (Python interface of the library was used). Each of such independent runs was performed until a stable iteration was reached.

Table 2. Community-aware node features used in our experiments. A combination of `WMD` and `CPC` is also used as a 2-dimensional embedding of a graph (`WMD+CPC`).

abbreviation	symbol	name	subsection
`CADA`	$\mathrm{cd}(v)$	anomaly score CADA	2.1
`CADA*`	$\overline{\mathrm{cd}}(v)$	normalized anomaly score	2.1
`WMD`	$z(v)$	normalized within-module degree	2.2
`CPC`	$p(v)$	participation coefficient	2.2
`CAS`	$\beta^*(v)$	community association strength	2.3
`CD_L11`	$L_1^1(v)$	L^1 norm for the 1st neighbourhood	2.4
`CD_L21`	$L_1^2(v)$	L^2 norm for the 1st neighbourhood	2.4
`CD_KL1`	$\mathrm{kl}_1(v)$	Kullback–Leibler divergence for the 1st neighbourhood	2.4
`CD_HD1`	$h_1(v)$	Hellinger distance for the 1st neighbourhood	2.4
`CD_L12`	$L_2^1(v)$	L^1 norm for the 2nd neighbourhood	2.4
`CD_L22`	$L_2^2(v)$	L^2 norm for the 2nd neighbourhood	2.4
`CD_KL2`	$\mathrm{kl}_2(v)$	Kullback–Leibler divergence for the 2nd neighbourhood	2.4
`CD_HD2`	$h_2(v)$	Hellinger distance for the 2nd neighbourhood	2.4

Classical (non-community-aware) node features are summarized in Table 3. These are standard and well-known node features. We omit their definitions but, instead, refer to the appropriate sources in the table; alternatively, see [17].

Finally, we will use two more sophisticated and powerful node features obtained through graph embeddings. Embeddings can be categorized into two main types: classical embeddings and structural embeddings. Classical embeddings focus on learning both local and global proximity of nodes, while structural embeddings learn information specifically about the local structure of nodes' neighbourhood. We test one embedding from each class: `node2vec` [13] and `struc2vec` [22].

3.3 Experiments

In this section, we present the results of two numerical experiments that were performed to investigate the usefulness of community-aware features:

1. *information overlap* between community-aware and classical features;
2. *combined variable importance for prediction* of community-aware and classical features.

Table 3. Classical (non-community-aware) node features that are used in our experiments.

abbreviation	name	reference
`lcc`	local clustering coefficient	[26]
`bc`	betweenness centrality	[11]
`cc`	closeness centrality	[2]
`dc`	degree centrality	[17]
`ndc`	average degree centrality of neighbours	[1]
`ec`	eigenvector centrality	[4]
`eccen`	node eccentricity	[6]
`core`	node coreness	[17]
`n2v`	16-dimensional `node2vec` embedding	[13]
`s2v`	16-dimensional `struc2vec` embedding	[22]

From the computational perspective, all analytical steps (generation of graphs, extractions of both community-aware and classical features, execution of experiments) were implemented in such a way that all experiments are fully reproducible. In particular, all steps that involve pseudo-random numbers were appropriately seeded. The source code allowing for reproduction of all results is available at GitHub repository[5].

Information Overlap In the first experiment, our goal was to test, using a variety of models, to what extent each community-aware feature described in Table 2 can be explained by all the classical features from Table 3. For each community-aware feature, we independently measured how well it is explained by each model via computing the Kendall correlation of the value of the selected feature and its prediction. To consider possible non-linear relationships, the non-parametric Kendall correlation was used that checks how well the ordering of predictions matches the ordering of the target. Nevertheless, we also used measures such as R^2, which assumes linearity and homoskedasticity of prediction error of the relationship, and the results were similar.

The tests were performed using 70/30 train-test split of data. To ensure that the reported results are robust, for each community-aware feature five models were built using random forest, xgboost, lightgbm, linear regression and, respectively, regularized regression. The maximum Kendall correlation that was obtained are reported.

The goal of this experiment is to show that community-aware features cannot be explained completely by classical features (including two highly expressible embeddings). The conclusion is that it is worth to include such features in predictive models as they could potentially improve their predictive power. However, this additional information could be simply noise and so not useful in practice. To verify the usefulness of the community-aware features, we performed

[5] https://github.com/sebkaz/BetaStar.git

two more experiments, namely, *one-way predictive power* and *combined variable importance for prediction* checks. In these experiments, we check if community-aware features are indeed useful in node label prediction problems. In the journal version of this paper the results of both experiments are reported. Here, due to space limitation, we only describe the results of *combined variable importance for prediction* experiment.

In general, for empirical graphs described in Section 3.1, the target is a binary feature that measures some practical feature or a role of a given node. It is important to highlight that these features are not derived from the community structure of these graphs, at least not directly. Instead, they are characteristics of nodes defined independently of the graph structure. Therefore, for these networks we do not expect that community-aware features will significantly outperform other features. However, we conjecture that in many empirical networks, it may be the case that the prediction target is related to the fact that a node is a strong member of its own community or not. We expect to see that some community-aware features are still useful in prediction. It is important to highlight that, as we have described in Section 3.1, we have not hand-picked a few empirical networks that present good performance of community-aware features, aiming for a diverse collection of networks.

Results and Observations For empirical graphs, in Table 4 we observe correlations significantly bounded away from 1. (For synthetic networks the correlations are even lower—see the journal version of this paper). In particular, for the **Grid** graph, the correlation values are the lowest in the family of the empirical graphs (slightly above 0.2 for single-community measures).

In summary, the presented results confirm that the information encapsulated in community-aware measures cannot be recovered completely using classical features (even including embeddings). In the following experiments, we investigate if this extra information is useful for the node classification task.

Combined Variable Importance for Prediction The second experiment (*combined variable importance for prediction*) provides a way to verify the usefulness of community-aware features for node classification task. For each graph we we build a single model predicting the target variable that takes into account all community-aware as well as all classical features (including both embeddings) as explanatory variables. A random forest classifier was built and the permutation variable importance [5,9] measure was computed for each feature using APS (average precision score) as a target predictive measure.

As in the previous experiment, a 70/30 train-test split was used. We report the ranking of variable importance (rank 1 being the most important one) so that the values are comparable across all graphs investigated in this experiment. The raw importance scores have different ranges for various graphs.

Results and Observations The results for empirical graphs are presented in Table 5. The ranks range between 1 and 53 (with rank 1 being the best), since

Table 4. Information overlap between community-aware and classical features. The maximum of Kendall correlation between target and predictions on test data set.

target	**Amazon**	**Facebook**	**Grid**	**LastFM**	**Reddit**
`CADA`	0.5830	0.5666	0.2156	0.4815	0.6826
`CADA*`	0.6058	0.5828	0.2174	0.5058	0.6867
`CPC`	0.6338	0.5992	0.2193	0.5175	0.7193
`CAS`	0.6538	0.6257	0.2999	0.5594	0.7306
`CD_L21`	0.7052	0.6464	0.3496	0.5698	0.7574
`CD_L22`	0.7554	0.7355	0.3557	0.6295	0.7941
`CD_L11`	0.7251	0.7041	0.6978	0.6220	0.7735
`CD_L12`	0.7794	0.7785	0.6447	0.6884	0.7810
`CD_KL1`	0.7176	0.7516	0.7394	0.6289	0.7755
`CD_HD1`	0.7383	0.7482	0.7168	0.6459	0.7853
`CD_KL2`	0.7706	0.7826	0.7292	0.6853	0.8097
`CD_HD2`	0.8212	0.8173	0.6930	0.7369	0.8221
`WMD`	0.8447	0.8456	0.8488	0.8531	0.7638

there are 53 features in total (13 community-aware, 8 classical, 16 for `node2vec`, and 16 for `struc2vec`). The rows are sorted by the arithmetic mean of rank correlations across all graphs.

For one empirical graph (namely, the *Facebook* graph), no community-aware measure appears in the top-10. It should be noted though, as can be seen by the experiments included in the journal version of this paper, that both `node2vec` and `struc2vec` embeddings provide almost perfect prediction for this graph. On the other hand, for the *Grid* graph, community-aware features are important (3 of them are in the top-10). In general, the community-aware features that score high for at least one graph are: `CAS`, `CD_L22`, `WMD`, `CD_L12`, `CD_HD2`, `CD_HD1`, and `CD_KL1`. In particular, we see that the second-neighbourhood measures are well represented. This indicates that looking at the community structure of larger ego-nets of nodes is useful for empirical graphs. This is not the case for synthetic **ABCD+o** graphs as their generation structure is simpler than the more sophisticated mechanisms that lead to network formation of empirical complex networks. (As always, we refer the reader to the journal version of this paper for experiments on synthetic networks.)

References

1. A. Barrat, M. Barthélemy, R. Pastor-Satorras, and A. Vespignani. The architecture of complex weighted networks. *Proceedings of National Academy of Sciences*, 101(11):3747–3752, 2004.
2. Alex Bavelas. Communication patterns in task-oriented groups. *The Journal of the Acoustical Society of America*, 22(6):725–730, 1950.
3. Smriti Bhagat, Graham Cormode, and S Muthukrishnan. Node classification in social networks. *Social network data analytics*, pages 115–148, 2011.
4. P. Bonacich and P. Lloyd. Eigenvector-like measures of centrality for asymmetric relations. *Social Networks*, 23:191–201, 2001.

Table 5. Variable importance ranks for community-aware features in models including all features as explanatory variables. Values range from 1 (the best) to 53 (the worst).

variable	**Amazon**	**Facebook**	**Grid**	**LastFM**	**Reddit**
CAS	16	17	6	6	40
CD_L21	19	32	11	29	25
CD_KL1	18	20	14	9	30
WMD	49	25	1	8	31
CADA	26	33	22	15	33
CPC	39	30	24	17	26
CD_L22	25	28	3	11	49
CD_L12	24	53	20	7	4
CADA*	37	34	26	14	50
CD_HD1	23	23	17	27	8
CD_L11	14	22	18	45	27
CD_KL2	15	31	27	42	28
CD_HD2	9	52	46	38	32

5. Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi:10.1023/A:1010933404324.
6. F. Buckley and F. Harary. *Distance in graphs (Vol. 2)*. Addison-Wesley, 1990.
7. Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006. URL: https://igraph.org.
8. Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S Yu. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM'20)*, 2020.
9. Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously, 2019. arXiv:1801.01489.
10. Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.
11. Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977. URL: http://www.jstor.org/stable/3033543.
12. Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.
13. Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. *CoRR*, abs/1607.00653, 2016. URL: http://arxiv.org/abs/1607.00653, arXiv:1607.00653.
14. Roger Guimera and Luís A Nunes Amaral. Functional cartography of complex metabolic networks. *nature*, 433(7028):895–900, 2005.
15. Thomas J. Helling, Jan C. Scholtes, and Frank W. Takes. A community-aware approach for identifying node anomalies in complex networks. In *International Workshop on Complex Networks & Their Applications*, 2018.
16. Thomas J Helling, Johannes C Scholtes, and Frank W Takes. A community-aware approach for identifying node anomalies in complex networks. In *Complex Networks and Their Applications VII: Volume 1 Proceedings The 7th International Confer-*

ence on Complex Networks and Their Applications COMPLEX NETWORKS 2018 7, pages 244–255. Springer, 2019.
17. Bogumił Kamiński, Paweł Prałat, and François Théberge. *Mining Complex Networks*. Chapman and Hall/CRC, 2021.
18. Bogumił Kamiński, Paweł Prałat, and François Théberge. Artificial benchmark for community detection with outliers (abcd+o). *Applied Network Science*, 8(1):25, 2023.
19. Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.
20. Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1269–1278, 2019.
21. Carsten Matke, Wided Medjroubi, and David Kleinhans. SciGRID - An Open Source Reference Model for the European Transmission Network (v0.2), Jul 2016. URL: http://www.scigrid.de.
22. Leonardo F.R. Ribeiro, Pedro H.P. Saverese, and Daniel R. Figueiredo. Struc2vec: Learning node representations from structural identity. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 385–394, 2017. doi:10.1145/3097983.3098061.
23. Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021.
24. Benedek Rozemberczki and Rik Sarkar. Characteristic Functions on Graphs: Birds of a Feather, from Statistical Descriptors to Parametric Models. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*, page 1325–1334. ACM, 2020.
25. Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):5233, 2019.
26. Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1988.
27. Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 974–983, 2018.